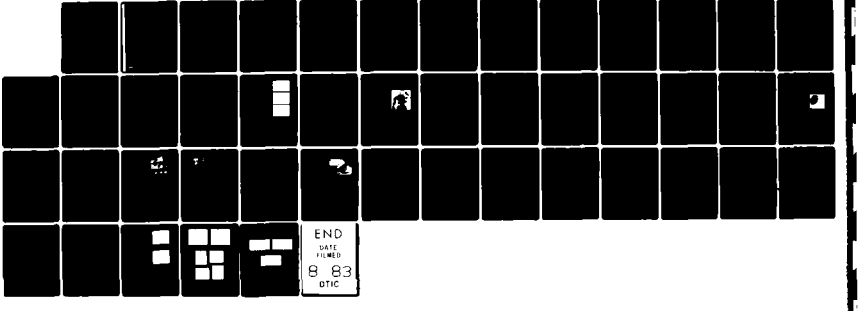
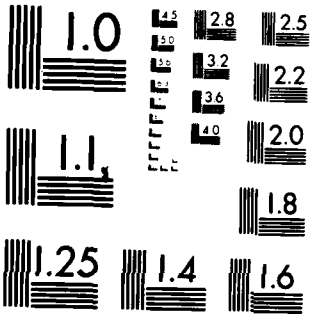
AD-A131 289 COMPUTER VISION RESEARCH AND ITS APPLICATIONS TO AUTOMATED CARTOGRAPHY(U) SRI INTERNATIONAL MENLO PARK CA M A FISCHLER 27 JUL 83 MDA903-83-C-0027 1/1

UNCLASSIFIED F/G 8/2 NL





(12)

COMPUTER VISION RESEARCH AND ITS APPLICATIONS TO AUTOMATED CARTOGRAPHY

First Semiannual Technical Report
Covering the period December 10, 1982 to June 10, 1983

Contract Amount:	$3,654,877
Effective Date:	December 10, 1982
Expiration Date:	September 30, 1985

July 27, 1983

By: Martin A. Fischler, Program Director
Principal Investigator, (415)859-5106

Artificial Intelligence Center
Computer Science and Technology Division

Prepared for:

Defense Advanced Research Projects Agency
1400 Wilson Boulevard
Arlington, Virginia 22209

Attention: Cdr. Ronald Ohlander, Program Manager
Information Processing Techniques Office

Contract No. MDA903-83-C-0027
DARPA Order No. 3862 and AMD 8
Program Code No. 3D30, Program Element 61101E
SRI Project 5355

SRI International
333 Ravenswood Avenue
Menlo Park, California 94025
(415) 326-6200
Cable: SRI INTL MPK
TWX: 910-373-2046

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER First Semiannual Technical Rpt.	2. GOVT ACCESSION NO. AD-A131	3. RECIPIENT'S CATALOG NUMBER 289
4. TITLE (and Subtitle) Computer Vision Research and Its Applications to Automated Cartography		5. TYPE OF REPORT & PERIOD COVERED Semiannual Technical 12/10/82 to 6/10/83
		6. PERFORMING ORG. REPORT NUMBER 5355 1st Semiannual Tech.
7. AUTHOR(s) Martin A. Fischler		8. CONTRACT OR GRANT NUMBER(s) MDA903-83-C-0027
9. PERFORMING ORGANIZATION NAME AND ADDRESS SRI International 333 Ravenswood Avenue Menlo Park, California 94025		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Program Code No. 3D30 Program Element 61101E
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency 1400 Wilson Boulevard Arlington, Virginia 22209		12. REPORT DATE July 27, 1983 13. NUMBER OF PAGES 44
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) DCASMA, San Francisco 1250 Bayhill Drive San Bruno, California 94066		15. SECURITY CLASS (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

image understanding, computer vision, automated cartography, feature extraction, stereo compilation, linear delineation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Our principal objective in this research program is to obtain solutions to fundamental problems in computer vision; particularly those problems that are relevant to the development of an automated capability for interpreting aerial imagery and the production of cartographic products.

Our plan is to advance the state of the art in selected core areas such as stereo compilation, feature extraction, linear delineation, and image

(CONTINUED ON NEXT PAGE)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

matching; also, to develop an "expert system" control structure which will allow a human operator to communicate with the computer at a problem oriented level, and guide the behavior of the low level interpretation algorithms doing detailed image analysis.

Finally, we plan to use the DARPA/DMA Testbed as a mechanism for transporting both our own and IU community advances, in image interpretation and scene analysis, to DMA, ETL, and other members of the user community.

A

CONTENTS

ABSTRACT

Our principal objective in this research program is to obtain solutions to fundamental problems in computer vision; particularly those problems that are relevant to the development of an automated capability for interpreting aerial imagery and the production of cartographic products.

Our plan is to advance the state of the art in selected core areas such as stereo compilation, feature extraction, linear delineation, and image matching; also, to develop an "expert system" control structure which will allow a human operator to communicate with the computer at a problem oriented level, and guide the behavior of the low level interpretation algorithms doing detailed image analysis.

Finally, we plan to use the DARPA/DMA Testbed as a mechanism for transporting both our own and IU community advances, in image interpretation and scene analysis, to DMA, ETL, and other members of the user community.

ACKNOWLEDGEMENT

The following researchers have contributed to the work described in this report: S. Barnard, R.C. Bolles, M.A. Fischler, M.J. Hannah, A.J. Hanson, D.L. Kashtan, K. Laws, A. Pentland, L.H. Quam, G.B. Smith, and H.C. Wolf.

I INTRODUCTION

A major focus of our current work is the construction of an Expert System for Stereo Compilation and Feature Extraction. Our intent in this effort is to develop a system that provides a framework for allowing higher level knowledge to guide the detailed interpretation of imaged data by autonomous scene analysis techniques. Such a system

would allow symbolic knowledge, provided by higher level knowledge sources, to automatically control the selection of appropriate algorithms, adjust their parameters, and apply them in the relevant portions of the image.

Recognizing the difficulty of completely automating the interpretation process, the expert system will be structured so that a human operator can provide the required high level information when there are no reliable techniques for automatically extracting this information from the available imagery. As new research results become available, the level of human interaction can be progressively reduced.

The expert system we are building can thus be viewed as an intelligent user-level interface for guiding semiautomated image processing activities. Such a system is envisioned as a rule-based system with a library of processes and activities, which can be invoked to carry out specific goals in the domain of cartographic analysis and stereo reconstruction. The system would depend on the human user for those types of information not easily extracted from the given imagery, and allow the computer system to take over in those areas where the utility of automated analysis has been clearly demonstrated.

Development of the expert system control structure is a research task still in an early stage of accomplishment. The remainder of this report will describe progress in research supporting the development of potential scene analysis components of the system, as well as other Image Understanding research of a more basic nature.

II RESEARCH PLANS AND PROGRESS

A. Development of Methods for Modeling and Using Physical Constraints in Image Interpretation.

Our goal in this work is to develop methods that will first allow us to produce a sketch of the physical nature of a scene and the illumination and imaging conditions, and next permit us to use this physical sketch to guide and constrain the more detailed descriptive processes -- such as precise stereo mapping.

Our approach is to develop models of the relationship between physical objects in the scene and the intensity patterns they produce in an image (e.g., models that allow us to classify intensity edges in an image as either shadow, or occlusion, or surface intersection, or material boundaries in the scene); models of the geometric constraints induced by the projective imaging process (e.g., models that allow us to determine the location and orientation of the camera that acquired the image, location of the vanishing points induced by the interaction between scene and camera, location of a ground plane, etc.); and models of the illumination and intensity transformations caused by the atmosphere, light reflecting from scene surfaces, and the film and digitization processes that result in the computer representation of the image.

These models, when instantiated for a given scene, provide us with the desired "physical" sketch. We are assembling a "constraint-based stereo system" that can use this physical sketch to resolve the ambiguities that defeat conventional approaches to stereo modeling of scenes (e.g., urban scenes or scenes of cultural sites) for which the images are widely separated in either space or time, or for which there are large featureless areas, or a significant number of occlusions.

Recent publications of our work in this area are cited in the references [1-4, 9-12]. Also see Appendicies A and B.

B. Stereo Compilation: Image Matching and Interpolation

We are implementing a complete state-of-the-art stereo system that produces dense range images from given pairs of intensity images. We plan to use this system both as a framework for our stereo research, and as the base component of our planned expert system.

There are five components of this stereo system: a rectifier, a sparse matcher, a dense matcher, an interpolator, and a projective display module. The rectifier estimates the parameters and distortions associated with the imaging process, the photographic process, and the digitization. These parameters are used to map digitized image coordinates onto an ideal image plane. The sparse matcher performs two-dimensional searches to find several matching points in the two images, which it uses to compute a relative camera model. The dense matcher tries to match as many points as possible in the two images. It uses the relative camera model to constrain the searches to one dimension, along epipolar lines. The interpolator computes a grid of range values by interpolating between the matches found by the dense matcher. The projective display module allows interactive examination of the computed 3-D model by generating 2-D projective views of the model from arbitrarily selected locations in space. Initial versions of all components of the system have been implemented.

Present research in this task is focused primarily on the image correspondence (matching) and interpolation problems. With respect to image matching, the following major issues are being addressed:

* What is a correct match?
* How does one measure the performance of a matcher?
* What causes existing matching techniques to fail?
* How can one improve the performance of matching techniques?

Since there are no reliable analysis techniques for evaluating the performance of matching algorithms when applied to real world images, we must evaluate them by extensive testing. To expedite such testing, a database of images and ideal match data (ground truth) is being

assembled. For example, we have acquired data from the ETL Phoenix test site that were produced specifically for testing matching techniques. Every point in the database we are constructing contains annotations that indicate the categories of matching problems for that point, and other information that might be useful to evaluate the performance or guide the application of matching techniques.

We are currently investigating a hypothesize - verify approach to local matching. Potential matches are verified by examining the image for compliance with the assumptions of the matching operator's model. For example, area correlation matching operators assume that correctly registered image patches will differ only by Gaussian noise. A simple verification technique is to examine the statistics of the point-by-point difference between the hypothesized alignment of the patches for conformance with that model. Image anomalies, such as moving objects or occluding contours, will typically produce a difference image that has a highly structured geometry, indicating the shape and location of the anomaly. Such anomalous areas can be removed from the region over which the correlation is computed, and the process iterates until either an acceptable match criterion is satisfied, or too many points are removed from the region.

In many cases (e.g., occlusion and featureless areas) local matching techniques are not capable of producing the required correspondences over regions of significant extent. We intend to use the information provided by the "physical sketch" (see previous section) to detect such situations, and to select alternative means for obtaining the required depth information.

As indicated above, when a stereo pair of images are matched, we generally can do no better than to compute a sparse depth map of the imaged scene. However, for many tasks a sparse depth map is inadequate. We want a complete model that accurately portrays the scene's surfaces. To achieve this goal, we must be able to obtain the missing surface shape information from the shading of the images of the stereo pair.

To understand the relationship between image shading and surface shape, we built a differential model [see references 10 and 11] that relates shape and shading but, unfortunately, does not provide a complete basis for a shape recovery algorithm [see reference 12]. However, the information available in image shading does allow the building of a surface interpolation algorithm that finds a surface that is consistent with the image shading. We are proceeding with such a development.

As image shading alone does not provide sufficient information to find surface orientation, further shape information sources in the image are needed. We are evaluating additional scene attributes that encode shape information in their image, and the models necessary to recover the corresponding shape information.

C. Feature Extraction: Scene Description, Partitioning, and Labeling

Our current research in this area addresses two related problems: (1) representing natural shapes such as mountains, vegetation, and clouds, and (2) computing such descriptions from image data. The first step towards solving these problems is to obtain a model of natural surface shapes.

A model of natural surfaces is extremely important because we face problems that seem impossible to address with standard descriptive computer vision techniques. How, for instance, should we describe the shape of leaves on a tree? Or grass? Or clouds? When we attempt to describe such common, natural shapes using standard shape-primitive representations, the result is an unrealistically complicated model of something that, viewed introspectively, seems very simple. Furthermore, how can we extract 3-D information from the image of a textured surface when we have no models that describe natural surfaces and how they evidence themselves in the image? The lack of such a 3-D model has restricted image texture descriptions to being ad hoc statistical measures of the image intensity surface.

Fractal functions, a novel class of naturally-arising functions, are a good choice for modeling natural surfaces because many basic physical processes (e.g., erosion and aggregation) produce a fractal surface shape, and because fractals are widely used as a graphics tool for generating natural-looking shapes. Additionally, we have recently conducted a survey of natural imagery and found that a fractal model of imaged 3-D surfaces furnishes an accurate description of both textured and shaded image regions, thus providing validation of this physics-derived model for both image texture and shading.

Encouraging progress relevant to computing 3-D information from imaged data has already been achieved by use of the fractal model. We have derived a test to determine whether or not the fractal model is valid for particular image data, developed an empirical method for computing surface roughness from image data, and made substantial progress in the areas of shape-from-texture and texture segmentation. Characterization of image texture by means of a fractal surface model has also shed considerable light on the physical basis for several of the texture partitioning techniques currently in use, and made it possible to describe image texture in a manner that is stable over transformations of scale and linear transforms of intensity.

The computation of a 3-D fractal-based representation from actual image data has been demonstrated. This work has shown the potential of a fractal-based representation for efficiently computing good 3-D representations for a variety of natural shapes, including such seemingly difficult cases as mountains, vegetation, and clouds.

This research is expected to contribute to the development of (1) a computational theory of vision applicable to natural surface shapes, (2) compact representations of shape useful for natural surfaces, and (3) real-time regeneration and display of natural scenes. We also anticipate adding significantly to our understanding of the way humans perceive natural scenes.

Details of this work can be found in Pentland [8], reproduced as Appendix C to this report.

D. Linear Delineation and Partitioning

A basic problem in machine vision research is how to produce a line sketch that adequately captures the semantic information present in an image. (For example, maps are stylized line sketches that depict restricted types of scene information.) Before we can hope to attack the problem of semantic interpretation, we must solve some open problems concerned with direct perception of line-like structure in an image and with decomposing complex networks of line-like structures into their primitive (coherent) components. Both of these problems have important practical as well as theoretical implications.

For example, the roads, rivers, and rail-lines in aerial images have a line-like appearance. Methods for detecting such structures must be general enough to deal with the wide variety of shapes they can assume in an image as they traverse natural terrain.

Most approaches to object recognition depend on using the information encoded in the geometric shape of the contours of the objects. When objects occlude or touch one another, decomposition of the merged contours is a critical step in interpretation.

We have recently made significant progress in both the delineation and the partitioning problems. Our work in delineation [5] is based on the discovery of a new perceptual primitive that is highly effective in locating line-like (as opposed to edge-like) structure.

Our work on decomposing linear structures into coherent components [see reference 6 and Appendix D] is based on the formulation of two general principles that appear to have applicability over a wide range of problems in machine perception. The first of these principles asserts that perceptual decisions must be stable under at least small perturbations of both the imaging conditions and the decision algorithm parameters. The second principle is the assertion that perception is an explanatory process: acceptable precepts must be associated with explanations that are both complete (i.e., they explain all the data) and believable (i.e., they are both concise and of limited complexity).

These new delineation and partitioning algorithms have produced excellent results in experimental tests on real data [see references 5 and 6 and Appendix D].

REFERENCES

1. S. Barnard and A. Pentland, "Three-Dimensional Shape from Line Drawings," Proceedings of the Image Understanding Workshop, Arlington, Virginia (June 1983) and IJCAI-83.

2. S. Barnard, "Methods for Interpreting Perspective Images," (in press) AI Journal (1983).

3. S. Barnard and M.A. Fischler, "Computational Stereo," ACM Computing Surveys, Vol. 14 (4) (December 1982).

4. M.A. Fischler, et.al., "Modeling and Using Physical Constraints in Scene Analysis," AAAI-82.

5. M.A. Fischler and H.C. Wolf "Linear Delineation," IEEE CVPR-83.

6. M.A. Fischler and R.C. Bolles "Perceptual Organization and Curve Partitioning," Proceedings of the Image Understanding Workshop, Arlington, Virginia (June 1983) and IEEE CVPR-83.

7. K. Laws, "On the Evaluation of Scene Analysis Algorithms," Proceedings of the Image Understanding Workshop, Arlington, Virginia (June 1983).

8. A. Pentland, "Fractal Based Description of Natural Scenes," Proceedings of the Image Understanding Workshop, Arlington, Virginia (June 1983) and IEEE CVPR-83.

9. A. Pentland, "Depth of Scene from Depth of Field," Proceedings of the Image Understanding Workshop (September 1982).

10. A. Pentland, "Local Analysis of the Image: Limitations and Uses of Shading," Proceedings of the (IEEE) Workshop on Computer Vision: Representation and Control, Rindge, New Hampshire (August 1982).

11. G.B. Smith, "The Recovery of Surface Orientation from Image Irradiance," Proceedings of the Image Understanding Workshop (September 1982).

12. G.B. Smith, "The Relationship Between Image Irradiance and Surface Orientation," Proceedings of the Image Understanding Workshop, Arlington, Virginia (June 1983) and IEEE CVPR-83.

Appendix A

THREE-DIMENSIONAL SHAPE FROM LINE DRAWINGS

By: Stephen T. Barnard and Alex P. Pentland

THREE-DIMENSIONAL SHAPE FROM LINE DRAWINGS

Stephen T. Barnard and Alex P. Pentland

SRI International, 333 Ravenswood Ave., Menlo Park, California 94025

ABSTRACT

The problem of interpreting the shape of a three-dimensional space curve from its two-dimensional perspective image contour is considered. Observation of human perception indicates that a good strategy is to segment the image contour in such a way as to obtain approximately planar segments. The orientation of the osculating plane (the plane in which the space curve lies) can then be estimated for these segments, and the three-dimensional shape recovered. The assumption of spatial isotropy is used to derive the theoretical results needed to formulate such an estimation strategy. The resulting estimation strategy allows a single three-dimensional structure (up to a single Necker reversal) to be assigned to any smooth image contour. An implementation is described and shown to produce an interpretation that is quite similar to the analytically correct one in the case of a helix, even though a helix has substantial torsion. The general applicability of the algorithm is discussed.

I Introduction

Much recent vision research has emphasized the importance of image contour for shape interpretation [1,2,3,4,5,6,7]. Tenenbaum and Barrow [1] argue that image contour, for example, is dominant over shape from shading. Pentland [8] has presented examples in which the addition of a contour substantially improved the interpretation of a shaded surface. It seems that contour is one of the strongest sources of information for shape perception.

One source of evidence of the strength of contour information is line drawings. When we examine a line drawing, our perception of the three-dimensional shape implied by such a drawing is nearly always clear and unambiguous. How can we account for this, given that purely geometrical constraints admit of an infinite number of valid interpretations?

A. An Observation About Human Perception

When we observe line drawings such as those in Figure 1 (a), we have a clear perception of a non-planar three-dimensional structure. Notice that if we were to segment each of these drawings at the circled points, each of the resulting segments would have the same shape as they did when they were still hooked together *and would be approximately planar*, as is shown in Figure 1 (b). Thus, for these line drawings the problem of recovering the three-dimensional structure can be reduced to the problems of (1) segmenting the curve into perceptually planar segments, and (2) finding the plane that contains each of the curve segments (the *osculating plane*) [9]. Once we know the orientation of the plane which contains a curve segment we can then easily determine its three-dimensional shape.

* The research reported herein was supported by the Defense Advanced Research Projects Agency under Contract No. MDA 903-83-C-0027; this contract is monitored by the U. S. Army Engineer Topographic Laboratory. Approved for public release, distribution unlimited.

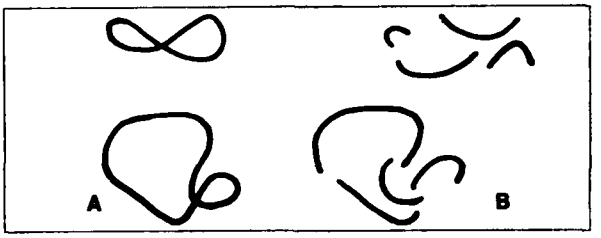


Figure 1. (a) Some Line Drawings, (b) Their Planar Subregions.

If we "by hand" try to segment image contours into planar regions, we find that the strategy can be successfully applied to a surprisingly large number of naturally-occurring image contours. For some contours, however, it is not obvious how well this strategy will work, primarily because there are no points which segment the space curve into planar regions. An example of such a curve is the helix shown in Figure 2 (a). Nonetheless, it may still be possible to obtain a good approximation of the three-dimensional *structure of such a curve using this strategy*.

B. A Strategy For Recovering Three-Dimensional Shape

This observation about human perception leads to the following processing strategy:

(1) Segment the image contour in such a way that each segment is likely to comprise a projection of a planar segment of the space curve.

(2) Calculate the planes implied by the segments from (1).

(3) Assemble the results of (2) into an estimate of the shape of the entire space curve.

The specific criteria for the initial segmentation are not dealt with here. It is clear, however, that the image contour should be segmented at singular points of curvature (maxima, minima, and inflection points). Hoffman and Richards [10] have presented a theory of curve segmentation that addresses this issue. Our approach will be to temporarily ignore the segmentation problem and to simply estimate the orientation of parts of the space curve from many local parts of the image contour. If valid results are forthcoming with this approach the method can only be improved with more elaborate segmentation.

C. Modeling the Space Curve

We shall model a space curve in the conventional way, as a three-dimensional vector function $\mathbf{x}(s)$ of one parameter s which is assumed to be a natural parameter, i.e., $|d\mathbf{x}(s)/ds| = 1$. The shape of such a curve is completely determined by two properties that are scalar functions of s: curvature, $\kappa(s)$, and torsion, $\tau(s)$ [9]. Curvature is always nonnegative; only straight lines and inflection points have zero curvature. Torsion may be intuitively defined as the amount of "twist" in the curve at a point s. Another way to visualize torsion is as the degree to which the osculating plane (the plane which contains the curve) is changing. Only planar

curves have zero torsion everywhere. Unlike curvature, torsion may be either negative or positive.

The presence of torsion is not directly evident in the image. It simply results in more or less foreshortening as the osculating plane of the contour varies. The effects of torsion, therefore, can be exactly mimicked by changes in curvature, and vice versa.

II Theory of Contour Interpretation

Not all three-dimensional interpretations of an image contour are equally likely. If we assume that spatial isotropy holds, then we know that viewer position is independent of the shape of the curve -- which allows us to make a reasonable guess about the latter's three-dimensional shape [8]. The first step towards a guess at the space curve's shape is the following proposition:

Proposition (Zero Torsion). The maximum-likelihood estimate of the torsion of the space curve is zero (i.e., no "twisting" of the curve).

This proposition follows because the assumption of spatial isotropy implies that the viewer's position and the shape of the space curve are mutually independent. Thus, not only is it unlikely that significant features of the curve will be hidden from view by coincidental alignment of the viewer and the curve, but, conversely, it is likely that the viewed scene will not change much with small changes in viewing position.* The appearance of a curve with substantial torsion** will change considerably with small changes in viewer position; if we assume spatial isotropy, therefore, we must expect that the torsion of the curve will be small.

Furthermore, given that spatial isotropy implies that the viewer position and the shape of the curve are mutually independent, the torsion of the curve must then also be independent of viewer position. Consequently, the torsion of the curve is as likely to be positive as negative, and thus the mean value (and maximum-likelihood estimate) for the magnitude of the torsion is zero† . The probability that the torsion is small implies this estimate will generally be a good one.

A. Estimation With The Assumption Of Zero Torsion

Even if we assume that torsion is zero (i.e., the space curve is planar), there is still a two-parameter set of space curves that could have generated that imaged contour. The two parameters correspond to the two degrees of freedom of the osculating plane.

Assume that we are given a small portion of an imaged contour, and asked to estimate the three-dimensional shape of the space curve which generated that image. If we measure the position and curvature at three points on the imaged contour, then we can uniquely define an elliptical arc that fits the image data. By the previous proposition, this elliptical arc is most likely caused by a space curve that is either an arc of a circle or of an ellipse, as those are the two planar (zero torsion) shapes which can project to an ellipse†† .

Previous research ([2], [12]) has shown that the maximum-likelihood estimate of the space curve's shape is given by the following proposition (see also [2]):

Proposition (Planar Interpretation). Given an elliptical segment of an image contour and that the space curve is planar, the maximum likelihood estimate of the space curve's three-dimensional shape is a segment of a circle.

Barnard [12] has constructed a maximum entropy estimator that implements this proposition for perspective images and that is tolerant of digitization noise. Operating under the assumption that the space curve has zero torsion, it chooses the orientation that maximizes the entropy of backprojected image contour curvature measurements. That is, curvature is first measured at several points in the image contour, then the curvatures of hypothetical planar space curves of essentially all orientations are computed by backprojection, and, finally, the orientation that leads to the space curve of most uniform curvature (in the sense of maximum entropy) is selected. In general, three image contour curvature measurements are sufficient for an unambiguous maximum-entropy interpretation (up to a Necker reversal).

III Three-Dimensional Estimation

Now let us return to the general problem of estimating the shape of the space curve, given a smooth imaged contour. Let us first take three curvature measurements along the imaged contour. These three measurements define an ellipse. As just described, this leads to a circular interpretation of the space curve. Now suppose that we have additional image contour curvature measurements. There are, then, two cases to consider:

First case: the new points fit on the same ellipse. In the first case we have quite strong evidence of the space curve's shape. For, if the osculating plane were changing, the curvature would have to be changing also — and in just such a manner as to exactly cancel (in the image) the effect of the changing osculating plane. Similarly, if the curvature of the space curve were changing, the osculating plane would have to change just exactly enough to cancel the effect of the changing curvature. As such a "conspiracy" to cancel the visible effects of change is unlikely (a direct violation of general position), we must conclude that there was neither torsion nor change in curvature, and, thus, there is a great (in fact, maximum) likelihood that the new image curvature measurements result from the same circular space curve defined by the first three measurements.

Second case: the new points don't fit on the same ellipse. What if the additional measurements lie off the ellipse defined by the first three measurements? Then we can be certain that either the curvature or the osculating plane (or both) of the space curve has changed. This new point is, therefore, a possible place to segment the curve. What we must do when we encounter such a point is advance along the image contour until we are completely past the point, and obtain a new estimate of the space curve's osculating plane. If the new osculating plane has the same orientation as the previous osculating plane, then we have evidence that the space curve continues to be planar, and we should not segment the curve. If, however, we obtain a different orientation for the osculating plane, then we should segment the space curve and begin a new planar segment of the curve.

As any smooth image contour may be closely approximated by portions of ellipses and straight lines* , this interpretation strategy will yield a single interpretation for the three-

*This is often referred to as the assumption of general position. Thus, spatial isotropy implies general viewing position.

**As a function of position on the image contour rather than as a function of s

†Note that at places where the curvature is zero — straight segments and inflection points — the torsion is not defined and may arbitrarily be taken to be zero. That is, the osculating plane may be changed freely at these points without affecting the shape of the space curve.

††This is true of both perspective and orthographic projection, however, we will deal exclusively with the more general case of perspective foreshortening.

*Only the third and higher derivatives of the imaged contour that will fail to be exactly matched. People, it should be noted, are very poor observers of changes in the third derivatives of an image contour.

dimensional shape of the space curve (up to Necker reversals). Further, this interpretation will be the most likely interpretation on a point-by-point basis. It should be noted that the first two steps of this estimation strategy are similar to the strategy proposed in [1].

IV An Example

The interpretation strategy has been implemented and applied to a synthetic image of a helical space curve. The helix example is a good test because a helix has significant torsion everywhere, thus, distinguished segmentation points do not exist and it is not clear what the estimation strategy will do. If we can recover the helical shape of the space curve with some accuracy, we shall have demonstrated that the estimation strategy can perform even when no good segmentation is available.

Figure 2 (a) shows a perspective image of a helix. Figure 2 (b) shows a plot of the spherical indicatrix of the helix. The spherical indicatrix is a plot of the orientation of the osculating plane of the space curve. The axes in this plot corresponds to the azimuth and elevation of the osculating plane. As mentioned previously, knowledge of the orientation (azimuth and elevation) of the osculating plane at each point, together with the imaged contour, uniquely determines the shape of the space curve. Thus, the spherical indicatrix is a method of displaying the three-dimensional shape of the space curve. Figure 2 (c) shows the spherical indicatrix estimated for the contour in (a). When this is compared with the actual indicatrix shown in (b), it is evident that the three-dimensional shape of the space curve has been fairly accurately recovered.

Summary. We have developed a theory for assigning a three-dimensional interpretation to any smooth image contour. The theory has been implemented and is undergoing evaluation, which may lead to further development. The results reported above indicate that the estimation strategy performs reasonably well even for cases such as a helix, where the presence of substantial torsion might have led one to expect poor performance.

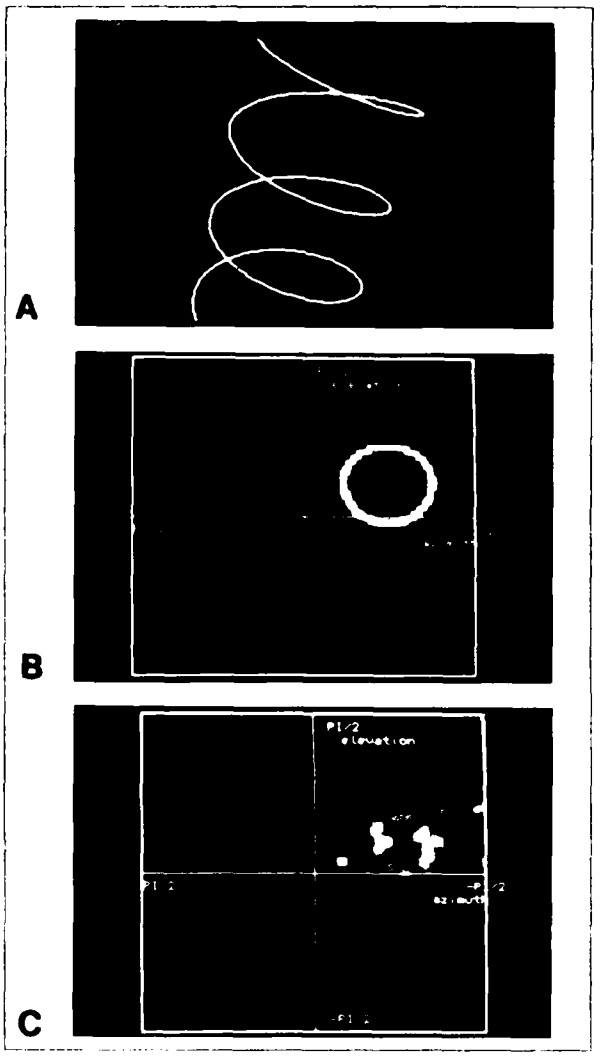


Figure 2. (a) An image of a helix, (b) the actual spherical indicatrix, (c) the recovered spherical indicatrix.

REFERENCES

[1] H. G. Barrow and J. M. Tenenbaum, Interpreting Line Drawings as Three-Dimensional Surfaces,*Artificial Intelligence* **17** (1981), 1-47.

[2] A. Witkin, Recovering Surface Shape And Orientation From Texture,*Artificial Intelligence* **17** (1981), 17-47.

[3] J. Kender, Shape From Texture,*Ph. D. Thesis, Computer Science Department, Carnegie-Mellon University* (1980).

[4] T. Kanade, Recovery of the 3-D Shape of an Object from a Single View,*Artificial Intelligence* **17** (1981), 409-460.

[5] K. A. Stevens, The Visual Interpretation of Surface Contours,*Artificial Intelligence* **17** (1981), 47-73.

[6] M. Brady and A. Yuille, An extremum principle for shape from contour,*In this proceedings* .

[7] D. G. Lowe and T. O. Binford, The Interpretation of Three-Dimensional Structure from Image Curves,*Proceedings of the 7th IJCAI, Aug. 24-28, University of British Columbia, Vancouver, B.C., Canada.* **2**, 613-618.

[8] A. Pentland, Local Inference of Shape: Computation From Local Features,*Ph.D. Thesis, Psychology Department, Massachusetts Institute of Technology* (1982).

[9] M. M. Lipshutz, *Differential Geometry*, McGraw-Hill, New York, New York, 1969.

[10] D. D. Hoffman and W. A. Richards, Representing Smooth Plane Curves for Recognition: Implications for Figure-Ground Reversal,*Proceedings of the AAAI, Aug. 18-20, 1982, Carnegie-Mellon University, Pittsburg, Pennsylvania* , 5-8.

[11] A. Witkin, Shape From Contour,*Ph.D. Thesis, Psychology Department, Massachusetts Institute of Technology* (1982).

[12] S. Barnard, Interpreting Perspective Images,*SRI Artificial Intelligence Center Technical Note 271* to appear in *Artificial Intelligence* (1982).

Appendix B

THE RELATIONSHIP BETWEEN IMAGE IRRADIANCE AND SURFACE ORIENTATION

By: Grahame B. Smith

THE RELATIONSHIP BETWEEN IMAGE IRRADIANCE AND SURFACE ORIENTATION

Grahame B. Smith

Artificial Intelligence Center, SRI International
Menlo Park, California 94025

ABSTRACT

A formulation of shape from shading is presented in which surface orientation is related to image irradiance without requiring detailed knowledge of either the scene illumination or the albedo of the surface material. The case for uniformly *diffuse reflection* and *perspective projection* is discussed in detail. Experiments aimed at using the formulation to recover surface orientation are presented and the difficulty of nonlocal computation discussed. We present an algorithm for reconstructing the 3-D surface shape once surface orientations are known.

1 INTRODUCTION

When the human visual system processes a single image, e.g., Figure 1, it returns a perceived 3-D model of the world, even when that image has limited contour and texture information. This 3-D model is underdetermined by the information in the 2-D image; the visual system has used the image data and its model of visual processing to reconstruct the 3-D world. While there are many information sources within the image, shading is an important source. Facial make-up or a cartoonist's shading, is an everyday example of the way shape, as perceived by our human visual system, is manipulated by shading information.

A primary goal of computer vision is to understand this process of reconstructing the 3-D world from 2-D image data, to discover the model, or models that allow 2-D data to infer 3-D structure. The focus of this work is the recovery of the 3-D orientation of surfaces from image shading.

We present a formulation of the shape-from-shading problem, i.e., recovering 3-D surface shape from image shading, that is derived under assumptions of perspective projection, *uniformly diffuse reflection*,[1] and constant reflectance. This formulation differs from previous approaches to the problem in that we neither make assumptions about the surface shape [2], nor use direct knowledge of the illumination conditions and the surface albedo [3]. The cost we incur for dispensing with these restrictions is the introduction of higher-order differentials into the equations relating surface orientation and image irradiance. The benefits we gain allow us to investigate the strength of the constraint imposed by shading upon shape. Past attempts to solve the shape-from-shading problem, as well as our own efforts, have been aimed at recovering surface shape from image patches for which the reflectance (albedo) can be considered constant.

Figure 1 Shape from Shading.

Previously we examined the influence exerted by the assumption of uniformly diffuse reflection [1], and indicated that the equations relating surface orientation to image irradiance could be expected to yield useful results even in cases in which the reflection is not uniformly diffuse. In that examination we assumed orthographic rather than perspective projection. A comparison of our previous work with this paper, however, shows that the structure of the formulation is not dependent upon the projection used.

If we add additional assumptions, e.g., constraints on the surface type, we can simplify the relationship between surface orientation and image irradiance. While it is not our goal to add constraints upon surface type, the assumption that the surface is locally spherical allows the approximate surface orientation to be recovered by local computation.

The research reported herein was supported by the Defense Advanced Research Projects Agency under Contract MDA903-83-C-0027 and by the National Aeronautics and Space Administration under Contract NASA 9-16664. These contracts are monitored by the U.S. Army Engineer Topographic Laboratory and by the Texas A&M Research Foundation for the Lyndon B. Johnson Space Center.

[1] We prefer the expression *isotropic scattering* to either *uniformly diffuse reflection*, or *Lambertian reflection*, as it emphasis that scene radiance is isotropic. However, uniformly diffuse reflection, and Lambertian reflection are the terms commonly used to indicate that the scene radiance is isotropic.

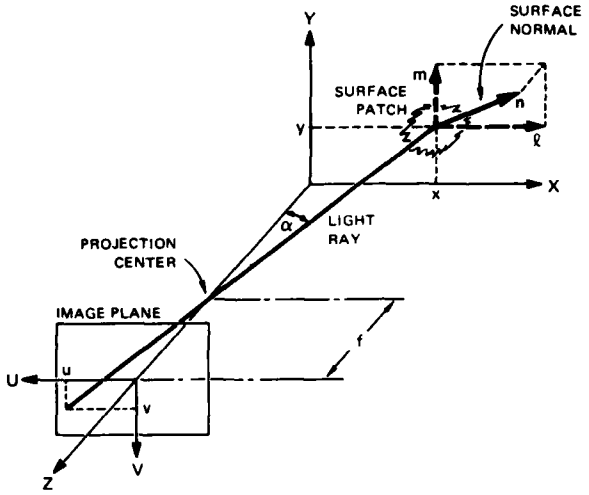


Figure 2 Coordinate Frame. X,Y,Z are the scene coordinates, U,V the image coordinates, and the image plane is located a distance f from the scene coordinate's origin - the projection center. α is the angle between the Z axis (the viewing direction) and the ray of light from the scene point (x, y, z) to the image point (u, v). l and m are the X and Y components of the surface normal n.

2 THE COORDINATE FRAME AND REPRESENTATION OF SURFACE ORIENTATION

The coordinate system we use is depicted in Figure 2. X,Y,Z are the scene coordinates and U,V are the image coordinates. The image and scene coordinates are aligned so that X and U axes are parallel, as are the Y and V axes. The U and V axes are inverted with respect to the X and Y axes, so that positive X and Y coordinates will correspond to positive U and V coordinates. The image plane is located at a distance f from the (perspective) projection center, the origin of the scene coordinates. A ray of light from the point (x, y, z) in the scene to the image point (u, v) makes an angle α with the viewing direction (i.e., the Z axis).

There are many parameterizations of the surface orientation: we choose to use (l, m), which are the X and Y components of the unit surface normal. In Figure 2, n is the unit normal of the surface patch located at (x, y, z); l and m are the components of this surface normal in the X and Y directions. From our viewing position we can see at most half the surfaces in the scene (i.e., those that face the viewer). The Z component of the surface normal has the magnitude $\sqrt{1 - l^2 - m^2}$, the sign determining whether the surface is forward-facing (has a positive Z component), or backward-facing (has a negative Z component). For large off-axis angle α, we see backward-facing surfaces near the edges of objects. The two components of the surface normal, l and m, do not provide an adequate parameterization of the surface in this case. Additionally, we need to know the sign of the Z component. Here we restrict ourselves to forward-facing surfaces. This minor restriction amounts to assuming that α is not too large and that we are not adjacent to an object's edge. Consequently, in this discussion we assume that the Z component of the surface normal is positive and that l and m constitute an adequate parameterization of scene surfaces.

3 IMAGE IRRADIANCE

The image irradiance equation we use is [4]

$$I(u, v) = R(l, m)\cos^4\alpha \quad ,$$

where $I(u, v)$ is the image irradiance as a function of the image coordinates u and v, and $R(l, m)$ is the surface radiance as a function of l and m, the components of the surface normal.[2] The term $\cos^4\alpha$ represents the off-axis effect of perspective projection. When α is small, $\cos^4\alpha$ is approximately unity; we then have the more familiar form of the image irradiance equation. From Figure 2 we see that

$$\cos\alpha = \frac{f}{\sqrt{u^2 + v^2 + f^2}} \quad .$$

Differentiating the image irradiance equation with respect to the image coordinates u and v, we obtain

$$\begin{aligned} I'_u &= R_l l_u + R_m m_u \quad , \\ I'_v &= R_l l_v + R_m m_v \quad , \end{aligned}$$

$$\begin{aligned} I'_{uu} &= R_{ll} {l_u}^2 + R_{mm} {m_u}^2 + 2R_{lm} l_u m_u + R_l l_{uu} + R_m m_{uu} \quad , \\ I'_{vv} &= R_{ll} {l_v}^2 + R_{mm} {m_v}^2 + 2R_{lm} l_v m_v + R_l l_{vv} + R_m m_{vv} \quad , \\ I'_{uv} &= R_{ll} l_u l_v + R_{mm} m_u m_v + R_{lm}(l_u m_v + l_v m_u) \\ &\qquad + R_l l_{uv} + R_m m_{uv} \quad , \end{aligned}$$

where subscripted variables denote partial differentation with respect to the subscript(s), and

$$I'_u = \left(\frac{1}{\cos^4\alpha}\right)\left(I_u + \frac{4uI}{u^2 + v^2 + f^2}\right) \quad .$$

$$I'_v = \left(\frac{1}{\cos^4\alpha}\right)\left(I_v + \frac{4vI}{u^2 + v^2 + f^2}\right) \quad .$$

$$I'_{uu} = \left(\frac{1}{\cos^4\alpha}\right)\left(I_{uu} + \frac{8uI_u}{u^2 + v^2 + f^2} + \frac{8u^2 I}{(u^2 + v^2 + f^2)^2} + \frac{4I}{u^2 + v^2 + f^2}\right) \quad .$$

$$I'_{vv} = \left(\frac{1}{\cos^4\alpha}\right)\left(I_{vv} + \frac{8vI_v}{u^2 + v^2 + f^2} + \frac{8v^2 I}{(u^2 + v^2 + f^2)^2} + \frac{4I}{u^2 + v^2 + f^2}\right) \quad .$$

$$I'_{uv} = \left(\frac{1}{\cos^4\alpha}\right)\left(I_{uv} + \frac{4vI_u}{u^2 + v^2 + f^2} + \frac{4uI_v}{u^2 + v^2 + f^2} + \frac{8uvI}{(u^2 + v^2 + f^2)^2}\right) \quad .$$

[2] Image irradiance is the light flux per unit area falling on the image, i.e., incident flux density. Scene radiance is the light flux per unit projected area per unit solid angle emitted from the scene, i.e., emitted flux density per unit solid angle.

If we are to use these expression to relate image measurements, e.g., I'_{uu}, to surface parameters l and m, then we must remove the derivatives of R.

4 UNIFORMLY DIFFUSE REFLECTION

To provide the additional constraints we need for relating *surface orientation to image irradiance*, we introduce constraints that relate properties of $R(l,m)$, — that is, constraints that specify the relationship between surface radiance and surface orientation. Such constraints are

$$(1-l^2)R_{ll} = (1-m^2)R_{mm} \quad ,$$
$$(R_{ll} - R_{mm})lm = (l^2 - m^2)R_{lm} \quad ,$$

where R_{ll} is the second partial derivative of R with respect to l, R_{mm} is the second partial derivative of R with respect to m, and R_{lm} is the second partial cross-derivative of R with respect to l and m.

These two partial differential equations embody the assumption of uniformly diffuse reflection. For uniformly diffuse reflection, $R(l,m)$ has the form

$$R(l,m) = al + bm + c\sqrt{1-l^2-m^2} + d \quad ,$$

where a, b, c, and d are constants, their values depending on illumination conditions and surface albedo. Note that l,m, and $\sqrt{1-l^2-m^2}$ are the components of the unit surface normal in the directions X,Y, and Z. $R(l,m)$ can be viewed as the dot product of the surface normal vector $(l, m, \sqrt{1-l^2-m^2})$ and a vector (a, b, c) denoting illumination conditions. As the value of a dot product is rotationally independent of the coordinate system, the scene radiance is independent of the *viewing direction* — which is the definition of uniformly diffuse reflection.

It is clearly evident that $R(l,m) = al + bm + c\sqrt{1-l^2-m^2} + d$ satisfies the pair of partial differential equations given above. In [1] we showed that $R(l,m) = al + bm + c\sqrt{1-l^2-m^2} + d$ *is the solution of the pair of partial differential* equations. These partial differential equations are an alternative definition of uniformly diffuse reflection.

It is worthy of note that $R(l,m) = al + bm + c\sqrt{1-l^2-m^2} + d$ includes radiance functions for multiple and extended illumination sources, including that for a hemispherical uniform source such as the sky. Of course, at a self-shadow edge R is not differentiable, so that the surfaces on each side of the self-shadow boundary have to be treated separately. The assumption of uniformly diffuse reflection restricts the class of material surfaces being considered, not the illumination conditions.

From the constraints for uniformly diffuse reflection, we derive the relationships

$$R_{ll} = \frac{1-m^2}{lm} R_{lm} \quad ,$$
$$R_{mm} = \frac{1-l^2}{lm} R_{lm} \quad .$$

Substituting these relationships for R_{ll} and R_{mm} in the expressions for I'_{uu},I'_{vv},and I'_{uv}, we obtain

$$[l_u^2(\frac{1-m^2}{lm}) + m_u^2(\frac{1-l^2}{lm}) + 2l_u m_u]R_{lm} = I'_{uu} - R_l l_{uu} - R_m m_{uu} \quad ,$$
$$[l_v^2(\frac{1-m^2}{lm}) + m_v^2(\frac{1-l^2}{lm}) + 2l_v m_v]R_{lm} = I'_{vv} - R_l l_{vv} - R_m m_{vv} \quad ,$$
$$[l_u l_v(\frac{1-m^2}{lm}) + m_u m_v(\frac{1-l^2}{lm}) + l_u m_v + l_v m_u]R_{lm} = I'_{uv} - R_l l_{uv} - R_m m_{uv} \quad .$$

By removing R_{lm} and substituting the expressions for R_l and R_m, defined by the expressions for I'_u and I'_v, we produce two partial differential equations relating surface orientation to image irradiance:

$$\alpha\theta l_{uu} + \beta\theta m_{uu} - \alpha\gamma l_{uv} - \beta\gamma m_{uv} = \chi\theta I'_{uu} - \chi\gamma I'_{uv} \quad ,$$
$$\alpha\theta l_{vv} + \beta\theta m_{vv} - \alpha\delta l_{uv} - \beta\delta m_{uv} = \chi\theta I'_{vv} - \chi\delta I'_{uv} \quad ,$$

where

$$\alpha = I'_u m_v - I'_v m_u \quad ,$$
$$\beta = I'_v l_u - I'_u l_v \quad ,$$
$$\gamma = l_u^2(1-m^2) + m_u^2(1-l^2) + 2l_u m_u lm \quad ,$$
$$\delta = l_v^2(1-m^2) + m_v^2(1-l^2) + 2l_v m_v lm \quad ,$$
$$\theta = l_u l_v(1-m^2) + m_u m_v(1-l^2) + (l_u m_v + l_v m_u)lm \quad ,$$
$$\chi = l_u m_v - l_v m_u \quad .$$

These equations relate surface orientation to image irradiance by parameter-free expressions. We make no assumptions about surface shape, nor do we need to know the parameters specifying illuminant direction, illuminant strength, and surface albedo. Our assumptions are about the properties of reflection in the world; these alone are sufficient to relate surface orientation to image irradiance. The above equations have been derived for the case of perspective projection; for orthographic projection, the primed ($'$) quantities are replaced by *their unprimed counterparts, e.g., I'_u is replaced by I_u*. The form of the equations is not a function of the projection used.

5 RECOVERY OF SURFACE ORIENTATION

It is difficult to solve the equations relating surface orientation to image irradiance, and thus to recover surface shape *from* observed image irradiance. We have used numerous integration schemes that characterize two distinct approaches. The two differential equations can be directly integrated in a step-by-step manner or, given some initial solution, a relaxation procedure may be employed. The difficulties that arise are twofold: numerical errors and multiple solutions.

Solutions of the equation $\chi = 0$ (the developable surfaces, e.g., a cylinder) are also solutions of the equations relating surface orientation to image irradiance. If the image intensities

were known in analytic form, the analytic approach to solving the equations could then employ boundary conditions to select the appropriate solution. However, since the analytic form for the image intensities is unknown, numerical procedures must be employed. The use of such procedures to directly integrate the equations inevitably introduces small errors. Such errors 'mix in' multiple solutions even when those solutions are incompatible with the boundary conditions. Instability of the numerical scheme seems responsible for the fact that such errors eventually dominate the recovered solution. A scheme that is representative of our various trials at direct integration is outlined.

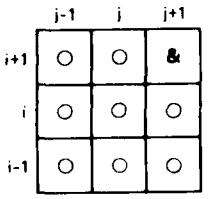
We transform our equations into finite-difference equations by using a three-point formula for the differentials of l and m. If $l(i,j)$ and $m(i,j)$ are the values of l and m at the (i,j)th pixel in the image, then at this pixel we use the finite-difference formulas,

$$l_u = \frac{l(i+1,j) - l(i-1,j)}{2} \quad ,$$

$$l_{uu} = l(i+1,j) + l(i-1,j) - 2l(i,j) \quad ,$$

$$l_{uv} = \frac{l(i+1,j+1) + l(i-1,j-1)}{4} - \frac{l(i+1,j-1) + l(i-1,j+1)}{4} \quad ,$$

and similar formulas for the other differentials. If we consider the 3 x 3 image patch centered on the (i,j)th pixel,



we could hope that the two finite difference equations, relating the eighteen values of l and m on the patch, could be solved explicitly for $l(i+1,j+1)$ and $m(i+1,j+1)$, (the (&) cell). Such a solution would allow l and m at the (&) cell to be calculated from the l's and m's at the (o) cells. Starting at some boundary at which we know l and m at the (o) cells, we can move along the image's row and then along the successive rows, calculating l and m at the (&) cell. However, examination of the surface-orientation-to-image-irradiance equations shows that we cannot solve these equations explicitly for l_{uv} and m_{uv} and that, consequently, we cannot obtain finite-difference equations that are explicit in the l and m of the (&) cell.

We avoid this difficulty by combining the two surface-orientation-to-image-irradiance equations into one and using surface continuity to provide the additional equation. Removing l_{uv} and m_{uv} from the differential equations, we have

$$\alpha(\delta l_{uu} - \gamma l_{vv}) + \beta(\delta m_{uu} - \gamma m_{vv}) = \lambda(\delta l'_{uu} - \gamma l'_{vv}) \quad .$$

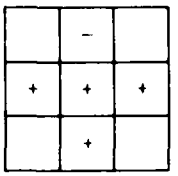
Surface continuity requires that $\frac{\partial^2 z}{\partial x \partial y} = \frac{\partial^2 z}{\partial y \partial x}$, from which it follows that

$$l_y(1-m^2) + m_y lm = m_x(1-l^2) + l_x lm \quad .$$

Provided that u and v are small compared with z (e.g., in the eye or in a standard-format camera), then

$$l_v(1-m^2) + m_v lm = m_u(1-l^2) + l_u lm \quad .$$

These two equations, which do not involve l_{uv} or m_{uv}, form a basis for finite difference equations that calculate l and m at the (-) cell from values of l and m at (+) cells.



The results obtained with the above integration scheme, together with many variations of it, are poor. Accurate values for l and m are obtained only within approximately five to ten rows of the known boundary. This is the case for noise-free image data. These results can be understood by examination of the finite-difference equations. The explicit expressions for l and m at the (-) cell are functions of the differences of l and m at the (+) cells. Such schemes are usually numerically unstable, making step-by-step integration impossible. While the failure to find a stable numerical scheme does not imply that one does not exist, our difficulty highlights the problem of finding numerical schemes, based on differential models, to propagate information from known boundaries. (One wonders whether nature experienced the same difficulties when designing the human vision system.)

Although the alternative to direct integration, a relaxation procedure to solve the equations, seems to offer relief from the numerical instability of direct integration, it nevertheless poses its own problems. The approach we used parallels the one in [3] for solving the image irradiance equation when the surface albedo and illumination conditions are known. For each image pixel we form three error terms: the residuals associated with the two surface-orientation-to-image-irradiance equations, and with the one surface continutiy equation. Minimizing the sum of the errors over the whole image with respect to l and m at each pixel produces an updating rule for l and m at each pixel. Given an initial solution, i.e., assignment of values for l and m at each pixel, a relaxtion scheme, like the one described, is useful only if it converges. While the constraint imposed by the underlying model is most important in ensuring convergence, the importance of a good initial solution for a relaxation method cannot be overemphasized. Simplifying the two partial differential equations (by using additional assumptions) provides a method for obtaining an good initial solution.

The spherical approximation assumes that we are viewing a spherical surface. This implies $l_y = 0, m_x = 0$, and $l_x = m_y$, namely, constant curvature that is independent of direction. Provided that u and v are small compared with z, then $l_v = 0, m_u = 0$ and $l_u = m_v$. For this case, the partial differential equations become relationships between image irradiance and its

derivatives, on the one hand, and the components of the surface normal, on the other:

$$\frac{1-m^2}{lm} = \frac{I'_{uu}}{I'_{uv}} \quad ,$$
$$\frac{1-l^2}{lm} = \frac{I'_{vv}}{I'_{uv}} \quad .$$

The spherical-approximation results for perspective projection are similar to those Pentland was able to obtain [2] for orthographic projection through local analysis of the surface. Besides providing a mechanism for obtaining an initial solution for a relaxation-style algorithm, they allow surface orientation to be estimated by purely local computation. Such an estimate will be exact when the surface is locally spherical.

The results of our experiments with relaxation procedures are easily summarized: the relaxation procedures were not convergent. While such nonconvergence is hardly unusual, the reasons for failure, however, are instructive. The residuals associated with both the surface-orientation-to-image-irradiance equations, and the surface continuity equations remain small during the relaxation, even when the solution is starting to diverge. Of course the residuals are not as small as they are when on the verge of solution, but they are small enough to make one believe that a solution has been obtained, particularly when the image is not noise-free. Apparently the equations are insensitive to particular values of l and m, being more concerned with the values of l_u, l_v, m_u, and m_v. As with direct integration, relaxation models need boundary conditions to select a particular solution. We used various boundary conditions in our relaxation experiments, but it is difficult to believe that a *model*, *apparently* insensitive to surface orientations, could be overly influenced by the surface orientations at a boundary.

Our two approaches, direct integration and relaxation, have not yielded a computational solution to the problem of recovering surface orientation from shading. The attractiveness of local computation is clear; it has neither numerical instability nor divergent behavior, but the cost it imposes is that assumptions must be made about surface shape. A compromise between some local computation and some information propagation may offer an approach that is not overly restrictive in its assumptions about surface shape. However, the question needs to be considered: Is the model underconstrained? Is shape recovery dependent on information other than shading? What other information (that is obtainable from the image), is necessary to enable the construction of effective shape-recovery algorithms?

6 RECONSTRUCTION OF THE SURFACE SHAPE

Surface orientation is not the same as surface shape. However, once we have obtained the surface orientation as a function of image coordinates, i.e., $l(u,v)$ and $m(u,v)$, we can use these to reconstruct the surface shape in the scene coordinates X,Y,Z. We derive a suitable formula.

Suppose we know the depth z_0 at scene coordinates (x_0, y_0, z_0), corresponding to (u_0, v_0) in the image. For the point $(x_0 + \Delta x, y_0 + \Delta y)$ we use the approximation

$$z(x_0+\Delta x, y_0+\Delta y) = z(x_0,y_0) + \Delta x \left.\frac{\partial z}{\partial x}\right|_{x_0,y_0} + \Delta y \left.\frac{\partial z}{\partial y}\right|_{x_0,y_0} .$$

Similarly,

$$z(x_1-\Delta x, y_1-\Delta y) = z(x_1,y_1) - \Delta x \left.\frac{\partial z}{\partial x}\right|_{x_1,y_1} - \Delta y \left.\frac{\partial z}{\partial y}\right|_{x_1,y_1} .$$

If $x_1 = x_0 + \Delta x$ and $y_1 = y_0 + \Delta y$, then

$$z(x_1,y_1) = z(x_0,y_0) + \frac{x_1-x_0}{2}\left(\left.\frac{\partial z}{\partial x}\right|_{x_0,y_0} + \left.\frac{\partial z}{\partial x}\right|_{x_1,y_1}\right) + \frac{y_1-y_0}{2}\left(\left.\frac{\partial z}{\partial y}\right|_{x_0,y_0} + \left.\frac{\partial z}{\partial y}\right|_{x_1,y_1}\right) .$$

Using the perspective transformation $u = -f\frac{x}{z}$ and $v = -f\frac{y}{z}$ to remove x and y, we obtain

$$z(u_1,v_1) = z(u_0,v_0) \times \frac{2f + u_0\left(\left.\frac{\partial z}{\partial x}\right|_{u_0,v_0} + \left.\frac{\partial z}{\partial x}\right|_{u_1,v_1}\right) + v_0\left(\left.\frac{\partial z}{\partial y}\right|_{u_0,v_0} + \left.\frac{\partial z}{\partial y}\right|_{u_1,v_1}\right)}{2f + u_1\left(\left.\frac{\partial z}{\partial x}\right|_{u_0,v_0} + \left.\frac{\partial z}{\partial x}\right|_{u_1,v_1}\right) + v_1\left(\left.\frac{\partial z}{\partial y}\right|_{u_0,v_0} + \left.\frac{\partial z}{\partial y}\right|_{u_1,v_1}\right)} .$$

As $\frac{\partial z}{\partial x} = \frac{-l}{\sqrt{1-l^2-m^2}}$ and $\frac{\partial z}{\partial y} = \frac{-m}{\sqrt{1-l^2-m^2}}$, we have the means of reconstructing the surface in scene coordinates from the values of surface orientation in image coordinates.

7 CONCLUSION

In this formulation of the shape-from-shading task, we have eliminated the need to know the explicit form of the scene radiance function by introducing higher-order derivatives into our model. This model is applicable to natural scenery without any additional assumptions about illumination conditions or the albedo of the surface material. However, without a computational scheme to reconstruct surface shap from image irradiance we may wonder if we have surrendered too much. The difficulties of finding a computational scheme must induce one to ask whether the model is underconstrained. Have we applied too few restrictions, thereby making shape recovery impossible? Notwithstanding the general concern about underconstraint of the model, the numerical difficulties encounted makes local computation of scene parameters attractive. Information propagation methods must always cope with the problem of accumulated errors. In our model, however, to achieve local computation we must make assumptions with regard to surface shape. What other information, besides shading, do we need to know if we are to recover surface shape? Can we find moderate restrictions that allow mostly local computation of the surface shape parameters? We are actively engaged in the pursuit of such procedures.

REFERENCES

1. Smith, G.B., From image irradiance to surface orientation, (submitted for publication). Available as Technical Note 273, Artificial Intelligence Center, SRI International (1982).
2. Pentland, A.P., The visual inference of shape: computation from local features, Ph.D. Thesis, Department of Psychology, Massachusetts Institute of Technology (1982).
3. Ikeuchi, K. and Horn, B.K.P., Numerical shape from shading and occluding boundaries, *Artificial Intelligence* 17 (1981) 141-184.
4. Horn, B.K.P. and Sjoberg R.W., Calculating the reflectance map, A.I. Memo 498, Artificial Intelligence Laboratory, Massachusetts Institute of Technology (1978).

Appendix C

FRACTAL-BASED DESCRIPTION OF NATURAL SCENES

By: Alex Pentland

FRACTAL-BASED DESCRIPTION OF NATURAL SCENES

Alex Pentland
Artificial Intelligence Center
SRI International
333 Ravenswood Ave., Menlo Park, CA. 94025

ABSTRACT

This paper addresses the problems of (1) representing natural shapes such as mountains, trees and clouds, and (2) computing such a description from image data. In order to solve these problems we must be able to relate natural surfaces to their images; this requires a good model of natural surface shapes. Fractal functions are good a choice for modeling natural surfaces because (1) many physical processes produce a fractal surface shape, (2) fractals are widely used as a graphics tool for generating natural-looking shapes, and (3) a survey of natural imagery has shown that the 3-D fractal surface model, transformed by the image formation process, furnishes an accurate description of both textured and shaded image regions. This characterization of image regions has been shown to be stable over transformations of scale and linear transforms of intensity.

Much work has been accomplished that is relevant to computing 3-D information from the image data, and the computation of a 3-D fractal-based representation from actual image data has been demonstrated using an image of a mountain. This example shows the potential of a fractal-based representation for efficiently computing good 3-D representations of natural shapes, including such seemingly-difficult cases as mountains, clumps of leaves and clouds.

Figure 1. Fractal-based models of natural shapes, by Mandelbrot and Voss [4].

1. INTRODUCTION

This paper addresses two related problems: (1) representing natural shapes such as mountains, trees and clouds, and (2) computing such a description from image data. The first step towards solving these problems, it appears, is to obtain a model of natural surface shapes. The task of finding such a model is extremely important to computer vision because we face problems that seem impossible to address with standard descriptive techniques. How, for instance, should we describe the shape of leaves on a tree? Or grass? Or clouds? When we attempt to describe such common, natural shapes using standard shape-primitive representations, the result is an unrealistically complicated model of something that, viewed introspectively, seems very simple.

Furthermore, how can we extract 3-D information from the image of a textured surface when we have no models that describe natural surfaces and how they evidence themselves in the image? The lack of such a 3-D model has generally restricted image texture descriptions to being *ad hoc* statistical measures of the image intensity surface. A good model of natural surfaces together with the physics of image formation would provide the analytical tools necessary for relating natural surfaces to their images. The ability to relate image to surface can provide the necessary leverage for dealing appropriately with the problems of finding a good representation for natural surfaces and computing such a description from the image data.

Even shape-from-shading [22,23] and surface-interpolation methods [24] are limited by the lack of a 3-D model of natural surfaces. Currently all such methods employ the heuristic of "smoothness" to relate neighboring points on the surface. Such heuristics are applicable to many man-made surfaces, of course, but are demonstrably untrue of most natural surfaces. In order to apply such techniques to natural surfaces, therefore, we must find a heuristic that is true of natural surfaces. Finding such a heuristic requires recourse to a 3-D model of natural surfaces.

* The research reported herein was supported by the Defense Advanced Research Projects Agency under Contract No. MDA 903-83-C-0027; this contract is monitored by the U. S. Army Engineer Topographic Laboratory. Approved for public release, distribution unlimited.

Fractal functions seem to provide such a model of natural surface shapes. Fractals are a novel class of naturally-arising functions, discovered primarily by Benoit Mandelbrot. Mandelbrot and others [1,2,4] have shown that fractals are found widely in nature and that a number of basic physical processes, such as erosion and aggregation, produce fractal surfaces. Because fractals look *natural* to human beings, much recent computer graphics research has focused on using fractal processes to simulate natural shapes and textures (see Figure 1), including mountains, clouds, water, plants, trees, and primitive animals [3,4,5,6,7]. Additionally, we have recently conducted a survey of natural imagery and found that a fractal model of imaged 3-D surfaces furnishes an accurate description of both textured and shaded image regions, thus providing validation of this physics-derived model for both image texture and shading [19].

2. FRACTALS AND THE FRACTAL MODEL

During the last twenty years, Benoit B. Mandelbrot has developed and popularized a relatively novel class of mathematical functions known as *fractals* [1,4]. Fractals are found widely in nature [1,2,4]. Mandelbrot shows that a number of basic physical processes, ranging from the aggregation of galaxies to the curdling of cheese, produce fractal surfaces. One general characterization is that any process that acts locally to produce a permanent change in shape will, after innumerable repetitions, result in a fractal surface. Examples are erosion, turbulent flow (e.g., of rivers or lava) and aggregation (e.g., galaxy formation, meteorite accretion, and snowflake growth). Fractals have also been widely and successfully used to generate realistic scenes (see Figure 1), including mountains, clouds, water, plants, trees, and primitive animals [3,4,5,6,7].

Perhaps the most familiar examples of naturally occurring fractal curves are coastlines. When we examine a coastline (as in Figure 1), we see a familiar scalloped curve formed by innumerable bays and peninsulas. If we then examine a finer-scale map of the same region, we shall again see the same type of curve. It turns out that this characteristic scalloping is present at *all* scales of examination [2], i.e., the statistics of the curve are invariant with respect to transformations of scale. This fact causes problems when we attempt to measure the length of the coastline, because it turns out that the length we are measuring depends not only on the coastline but also on the length of the measurement tool itself [2]! This is because, whatever the size measuring tool selected, all of the curve length attributable to features smaller than the size of the measuring tool will be missed. Mandelbrot pointed out that, if we generalize the notion of dimension to include *fractional* dimensions (from which we get the word "fractal"), we can obtain a consistent measurement of the coastline's length.

The definition. A fractal is defined as a set for which the Hausdorff-Besicovich dimension is strictly larger than the topological dimension. Topological dimension corresponds to the standard, intuitive definition of "dimension." Hausdorff-Besicovich dimension D, also referred to as the *fractal dimension*, may be illustrated (and roughly defined) by the examples (1) of measuring the length of an island's coastline, and (2) measuring the area of the island.

To measure the length of the coastline we might select a *measuring stick* of length λ and determine that n such measuring sticks could be placed end to end along the coastline. The length of the coastline is then intuitively $n\lambda$. If we were measuring the area of the island, we could use a square of area λ^2 to derive an area of $m\lambda^2$, where m is the number of squares it takes to cover the island. If we actually did this, we would find that both of these measurements vary with λ, the length of the measuring instrument -- an undesirable result.

In these two examples the length λ is raised to a particular power: the power of one to measure length, the power of two to measure area. These are two examples of the general rule of raising λ to a power that is the *dimension* of the object being measured. In the case of the island, raising λ to the topological dimension does not yield consistent results. If, however, we were to use the power 1.2 instead of 1.0 to measure the length, and 2.1 instead of 2.0 to measure the area, we would find that the measured length and area remained constant regardless of the size of the measuring instrument chosen.* The positive real number D that yields such a consistent measurement is the *fractal dimension*. D is always greater than or equal to the topological dimension.

The most important lesson the work of Mandelbrot and others teaches us is the following:

Standard notions of length and area do not produce consistent measurements for many natural shapes: the basic metric properties of these shapes vary as a function of the fractal dimension. Fractal dimension, therefore, is a *necessary* part of any consistent description of such shapes.

This result, which could almost be stated as a theorem, demonstrates the *fundamental* importance of knowing the fractal dimension of a surface. It implies that *any* description of a natural shape that does not include the fractal dimension cannot be relied upon to be correct at more than one scale of examination.

Fractal Brownian functions. Virtually all the fractals encountered in physical models have two additional properties: (1) each segment is statistically similar to all others; (2) they are statistically invariant over wide transformations of scale. Motion of a particle undergoing Brownian motion is the canonical example of this type of fractal. The discussion that follows will be devoted exclusively to fractal Brownian functions, a generalization of Brownian motion.

A random function $B(x)$ is a fractal Brownian function if for all x and Δx

$$Pr\left(\frac{B(x+\Delta x)-B(x)}{|\Delta x|^H} < y\right) = F(y) \qquad (1)$$

where $F(y)$ is a cumulative distribution function [1]. The fractal

*This example is discussed at greater length in Mandelbrot's book, "Fractals: Form, Chance and Dimension." The empirical data are from Richardson 1961.

dimension D of the graph described by $B(x)$ is

$$D = 2 - H \tag{2}$$

If $H = 1/2$ and $F(y)$ is a zero-mean Gaussian with unit variance, then $B(x)$ is the classical Brownian function. This definition has obvious extensions to two or more topological dimensions. The fractal dimension of a fractal Brownian function can also be measured from its Fourier power spectrum, as the spectral density of a fractal Brownian function is proportional to f^{-2H-1}. Discussion of the rather technical proof of this fact may be found in [1].

The fractal dimension of a surface corresponds roughly to our intuitive notion of jaggedness. Thus, if we were to generate a series of scenes with the same 3-D relief but increasing fractal dimension D, we would obtain the following sequence: first, a flat plane ($D \approx 2$), then rolling countryside ($D \approx 2.1$), a worn, old mountain range ($D \approx 2.3$), a young, rugged mountain range ($D \approx 2.5$), and finally a stalagmite-covered plane ($D \approx 2.8$).

The fractal dimension of a surface is invariant with respect to transformations of scale, as Δx is independent of H and $F(y)$. The fractal dimension is also invariant with respect to linear transformations of the data and thus it remains stable over smooth, monotonic transformations.

2.1 Fractals And The Imaging Process

Before we can use a fractal model of natural surfaces to help us understand images, however, we must determine how the imaging process maps a fractal surface shape into an image intensity surface. The mathematics of this problem is difficult and no complete solution has as yet been achieved. Nonetheless, simulation of the imaging process with a variety of fractal surface models can provide us with an empirical answer -- i.e., that images of fractal surfaces are themselves fractal as long as the fractal-generating function is spatially isotropic [19]. It is worth noting that practical fractal-generation techniques, such as those used in computer graphics, have had to constrain the fractal generating function to be isotropic so that realistic imagery could be obtained [3].

Real images do not, of course, appear fractal over all possible scales of examination. The overall size of the imaged surface places an upper limit on the range of scales for which the surface shape appears to be fractal, and a lower limit is set by the size of the surface's constituent particles. In between these limits, however, we may use Equation (1) to obtain a useful description of the surface.

Simulation shows that the fractal dimension of the physical surface dictates the fractal dimension of the image intensity surface; it appears that the fractal dimension of the image is a logarithmic function of the fractal dimension of the surface. If we assume that the surface is homogeneous, therefore, we can estimate the fractal dimension of the surface by measuring the fractal dimension of the image data. Even if the surface is not homogeneous, we can still infer the fractal dimension of the surface from imaged surface contours and bounding contours, by use of Mandelbrot's results.

What we have developed, then, is a method for inferring a basic property of the 3-D surface (its fractal dimension) from the image data. The fact that the fractal dimension corresponds closely to our intuitive notion of roughness shows the importance of the measurement: we can now discover from the image data whether the 3-D surface is rough or smooth, isotropic or anisotropic. We can know, in effect, what kind of cloth the surface was cut from. The fact that the fractal dimension also describes the basic metric properties of the imaged surface is further indication that it is a critical element in any consistent representation of natural surfaces.

2.2 Applicability Of The Fractal Model

An implication of the fractal surface model is that the image intensity surface is itself fractal -- and *vice versa*. This is because image intensity is primarily a function of the angle between the surface normal and the incident illumination; thus, if the image intensities satisfy Equation (1), then (for a homogeneous surface) the angle between surface normal and illuminant must also and, integrating, we find that the 3-D surface is a spatially isotropic fractal.

A method of evaluating the usefulness of the fractal surface model, therefore, is to determine whether or not images of natural surfaces are well described by a fractal function. To evaluate the applicability of the fractal model, we first rewrite Equation (1) to obtain the following description of the manner in which the second-order statistics of the image change with scale:

$$E(|dI_{\Delta x}|)\|\Delta x\|^{-H} = E(|dI_1|) \tag{3}$$

where k is a constant and $E(dI_{\Delta x})$ is the expected value of the change in intensity over distance Δx. Equation (3) is a hypothesized relation among the image intensities; a hypothesis that we may test statistically. If we find that Equation (3) is true of the image intensity surface and the viewed surface is homogeneous and continuous then we may conclude that the 3-D surface is itself fractal. It is an important characteristic of the fractal model that we can determine its appropriateness for particular image data because it means that we can know when, and when *not*, to use the model.

To evaluate the suitability of a fractal model for natural textures, the homogeneous regions from each of six images of natural scenes were densely sampled. In addition, twelve textures taken from Brodatz [8] were digitized and examined (see Figure 3). The intensity values within each of these regions were then approximated by a fractal Brownian function and the approximation error observed.

For the majority of the textures examined (77%), the model described the image data accurately (see [19] for more detail). In 15% of the cases the region was constant except for random, zero-mean perturbations; consequently, the fractal function correctly approximates the image data, although the fractal dimension was equal to the topological dimension and thus the data's dimension is technically not "fractional." The fit was poor in only 8% of the regions examined and, in many of these cases, it appeared that the image digitization had become saturated.

The fact that the vast majority of the regions examined were quite well approximated by a fractal Brownian function indicates that the fractal surface model will provide a useful description of natural surfaces and their images. Fractal Brownian functions

do *not*, of course, account for such large-scale spatial structure as those seen in the image of a brick wall or a tiled floor. Such structures must be accounted for by other means.

3. INFERRING SURFACE PROPERTIES

Fractal functions appear to provide a good description of natural surface textures and their images; thus, it is natural to use the fractal model for texture segmentation, classification and shape-from-texture. The first four headings of this section describe the research that has been performed in this area, and indicate likely directions for further research.

Fractal functions with $H \approx 0$ can be used to model smooth surfaces and their reflectance properties. For the first time, therefore, we can offer a single model encompassing both image shading and texture, with shading as a limiting case in the spectrum of texture granularity. The fractal model thus allows us to make a reasonable and rigorous definition of the categories "texture" and "shading," thus enabling us to discover similarities and differences between them. The final heading of this section briefly discusses this result.

3.1 An Example Of Texture Segmentation

Figure 2(a) shows an aerial view of San Francisco Bay. This image was digitized and the fractal dimension computed for each 8×8 block of pixels. Figure 2(b) shows a histogram of the fractal dimensions computed over the whole image. This histogram of fractal dimension was then broken at the "valleys" between the modes of the histogram, and the image segmented into pixel neighborhoods belonging to one mode or another.* Figure 2(c) shows the segmentation obtained by thresholding at the breakpoint indicated by the arrow under (b); each pixel in (c) corresponds to an 8×8 block of pixels in the original image. As can be seen, a good segmentation into water and land was achieved - one that cannot be obtained by thresholding on image intensity.

This image was then averaged down, from 512×512 pixels into 256×256 and 128×128 pixel images, and the fractal dimension recomputed for each of the reduced images. Figures 4 (d) and (e) illustrate the segmentations produced by using the *same* breakpoint as had been employed in the original full-resolution segmentation. These results demonstrate the stability of the fractal dimension measure across wide (4 : 1) variations in scale.

Several other images have been segmented in this manner [19]. In each case a good segmentation was achieved. The computed fractal dimension, and thus the segmentation, was found to be stable over at least 4 : 1 variations in scale; most were stable over a range of 8 : 1. Stability of the fractal description is to be expected, because the fractal dimension of the image is directly related to the fractal dimension of the viewed surface,

*No attempt was made to incorporate orientational information into measurement of the local fractal dimension, i.e., differences in dimension among various image directions at a point were collapsed into one average measurement.

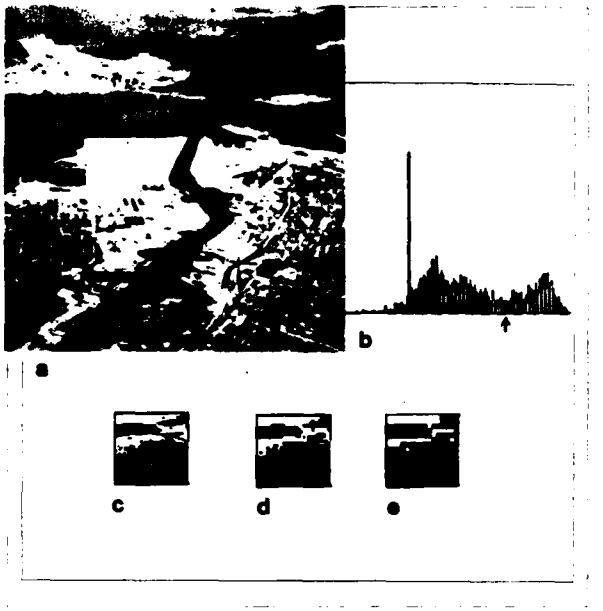


Figure 2. San Francisco Bay, and its texture segmentations.

which is a property of natural surfaces that has been shown to be invariant with respect to transformations of scale [2].

The fact that the fractal description of texture is stable with respect to scale is a critically important property. After all, consider: **how can we hope to compute a stable, viewer-independent representation of the world if our information about the world is not stable with respect to scale?** This example of texture property measurement reiterates what we observed earlier, i.e., the fact that the fractal dimension of the surface is *necessary* to any consistent description of a natural surface.

3.2 A Comparison With Other Segmentation Techniques

To obtain an objective comparison with previously established texture segmentation techniques, a mosaic of eight natural textures taken from Brodatz [8] was redigitized. The digitized texture mosaic, shown in Figure 3, was constructed by Laws [9,10] for the purpose of comparing various texture segmentation procedures. The textures that comprise this data set were chosen to be as visually similar as possible; gross statistical differences were removed by mean-value- and histogram-equalization.

Segmentation performance for these data exists for several techniques and, although differences in digitization complicate any comparisons we might wish to make, Laws's performance figures nevertheless serve as a useful yardstick for assessing performance on this data.

For this comparison simple orientational information was incorporated into the fractal description; the fractal dimension was calculated separately for the x and y coordinates. The two-parameter fractal segmenter yielded a theoretical classification accuracy of 84.4%. This compares quite favorably with correlation techniques [11,12] reported by Laws as attaining 65% ac-

Figure 3. The Brodatz textures used for comparison.

curacy, as well as with co-occurrence techniques [13,14] reported to be 72% accurate. This superior performance was achieved despite the large number of texture features employed by the other methods.

The simple two-parameter fractal segmenter even compares well with Laws's own texture energy statistics; even though his segmentation procedure included more than a dozen texture statistics that were optimized for the test data, its theoretical segmentation accuracy was only 3% better. Thus, the results of this comparison indicate that fractal-based texture segmentation will likely prove to be a general and powerful technique (for more details, see [19]).

3.3 Relationship To Texture Models

The fact that the fractal dimension of the image data can be measured by using either co-occurrence statistics in conjunction with Equation (1), or by means of the Fourier power spectrum, suggests one interesting aspect of the fractal model: it highlights a formal link between co-occurrence texture measures [13,14] and Fourier techniques [15,16,17]. The mathematical results Mandelbrot derives for fractal Brownian functions show that the way interpixel differences change with distance determines the rate at which the Fourier power spectrum falls off as frequency is increased, and vice versa.

Thus, it appears that the fractal model offers potential for unifying and simplifying the co-occurrence and Fourier texture descriptions. If we believe that natural surface textures and their images are fractal (as seems to be indicated by the previous results), then the fractal dimension is the most relevant parameter in differentiating among textures. In this case we would expect both the Fourier and co-occurrence techniques to provide reasonable texture segmentations, as both yield sufficient information to determine the fractal dimension. The advantage of the fractal model would be that it captures a simple physical relationship underlying the texture structure — a relationship lost with either of the other two characterizations of texture. Knowledge of the fundamental physical principle can result in both increased computational efficiency and further insight.

3.4 Shape From Texture

There are two ways surface shape is reflected in image texture: (1) projection foreshortening, a function of the angle between the viewer and the surface normal, and (2) the perspective texture gradient that is due to increasing distance between the viewer and the surface. These two phenomena are independent in that they have separate causes. Thus, they can serve to confirm each other — i.e., if projection foreshortening is used to estimate surface tilt, that estimate is ***independently confirmed*** if there is a texture gradient of the proper magnitude and same direction [17,18]. We may be confident our estimate is correct when such independent confirmation is found.

The fractal dimension found in the image appears to be nearly independent of the orientation of the surface (by virtue of independence with respect to scale); therefore fractal dimension cannot be used to measure surface orientation. Projection foreshortening does, however, affect the variance of the distribution $F(y)$ associated with the fractal dimension (see Equation (1)). Foreshortening affects $Var(F(y))$ in exactly the manner it affects the distribution of tangent direction.

Thus, to estimate surface orientation, we might assume that the surface texture is isotropic and estimate surface orientation on the basis of previously derived results [18]. While this often works [19], the necessity of assuming isotropy is a serious shortcoming of this technique. An important new result, therefore, is that we may in part cure this problem by observing the fractal dimensions in the x and y directions. If they are unequal we have *prima facie* evidence of anisotropy in the surface texture, because fractal dimension is unaffected by projection.

However a foreshortening-derived estimate of surface orientation is produced, we may still seek *confirmation* of it by measuring the perspective texture gradient; if confirmation is found, we may be confident of our estimate. Such a gradient appears in Figure 2: the houses dwindle in size with increasing distance from the viewer. Initial results, detailed in [19], indicate that perspective texture gradients can be inferred from the locally computed fractal dimension.

This two new results, i.e., the ability to obtain evidence of surface texture anisotropy and the measurement of the perspective texture gradient, are extremely important because they offer a way to make shape-from-unfamiliar-texture techniques sufficiently reliable so as to be useful. Development of these techniques, therefore, constitute an important task for future research.

3.5 Shading And Texture

Fractal functions with $H \approx 0$ can be used to model smooth surfaces and their reflectance properties accurately. When $H \approx 0$, the surface is locally planar, except for small, random variations described by the function $F(y)$ in Equation (1). If we assume that incident light is reflected at the angle of incidence and we make the variance of $F(y)$ small relative to the pixel size, the surface will be mirrorlike. If, on the other hand, the variance of $F(y)$ is large relative to the pixel size, the surface will become more Lambertian.

The fractal model, therefore, is a single model that can account for both image shading and texture, with shading corresponding to the limiting value of H. The fractal model thus allows us to make a reasonable and rigorous definition of the categories "texture" and "shading," in terms that can be measured by using the image data. One important goal of future research

will be to discover similarities or differences between these two categories; initial results indicate that local shape-from-shading results [26] can be generalized to include shape-from-texture.

4. COMPUTING A DESCRIPTION

Current methods for representing the three-dimensional world suffer from a certain awkwardness and inflexibility that makes them difficult to envisage as the basis for human-performance-level capabilities. They have encountered problems in dealing with partial knowledge or uncertain information, and they become implausibly complex when confronted with the problem of representing a crumpled newspaper, a clump of leaves or a puffy cloud. Furthermore, they seem ill-suited to solving the problem of representing a *class* of objects, or determining that a particular object is a member of that class.

What is wrong with conventional shape representations? One major problem is that they make too much information explicit. Experiments in human perception [21] lead one to believe that our representation of a crumpled newspaper (for instance) is not accurate enough to recover every z value; rather, it seems that we remember the general "crumpledness" and a few of the major features, such as the general outline. The rest of the newspaper's detailed structure is ignored; it is unimportant, *random*.

From the point of view of constructing a representation, the only important constraints on shape are the crumpledness and general outline. What we would like to do is somehow capture the notion of **constrained chance**, that is, the intuition that "a crumpled newspaper has x, y and z structural regularities and the rest is just variable detail," thus allowing us to avoid dealing with inconsequential (random) variations and to reason instead only about the structural regularities.

4.1 The Process Of Computing A Description

How shall we go about computing such a "constrained chance" description?* Let us consider the problem formally and see where that leads us. The process of computing a shape description (given some sensory data) seems best characterized as attempting to confirm or deny such hypotheses as "shape x is consistent with these sense data." Computation of a shape description, therefore, seems to be a problem in induction [20].

If, naively, we try to use an inductive method, we start with the set of all possible shape hypotheses; we then attempt to winnow the set down to a small number of hypotheses that are confirmed by the sensory data. The "set of all shape hypotheses," however, is much too large to work with. Consequently, we must take a slightly different tack.

Using the notion of constrained chance. Rather than attempting to enumerate "all shape hypotheses" explicitly, let us instead construct a *shape generator* that uses a random number generator to produce a surface shape description (I shall shortly describe how to do this). If we were to run this shape generator for an infinite period, it would eventually produce instances of every shape within a large class of shapes. If the generator were so constructed that the class of shapes produced was exactly the set of "all hypotheses" about shape, then the program for the shape generator, together with a the program for the random number generator, would comprise a description of the set of all shape hypotheses.

The shape generator illustrates how the notion of constrained chance may be used to obtain a compact description of an infinite set of shapes. By changing the constraints that determine how the output of the random number generator is translated into shape, we can change the set of shapes described; specifically, we can introduce constraints that rule out some classes of shape and thus restrict the set of shapes that are described. The ability to progressively restrict the set of shapes described allows us to use the constrained-chance shape generator as the basis for induction, rather than being forced to use the explicitly enumerated set of all shape hypotheses.

The process of computing a "constrained chance description" is straightforward. We use image data to infer (using knowledge of the physics of image formation) constraints on the shape, and then introduce those constraints into the shape generator. The end result will be a programlike description that is capable of producing all the shapes that are consistent with the image data; i.e., we shall have a description of the shapes confirmed by the image data. This, then, is the type of description we wanted: a description of shape that contains the important structural regularities that can be inferred from the image (e.g., crumpledness, outline), but one that leaves everything else as variable, random.

Some people are already doing this. Something very much like this constrained-chance representation is already being widely utilized in the computer graphics community. Natural-looking shapes are produced by a simple fractal program that recursively subdivides the region to be filled, introducing random jaggedness of appropriate magnitude at each step [3,5]. The jaggedness is determined by specifying the fractal dimension. The shapes that can be produced in this manner range from planar surfaces to mountainlike shapes, depending on the fractal dimension. Current graphics technology often employs fractal shape generators in a more constrained mode; often the overall, general shape or the boundary conditions are specified beforehand. Thus, a scene is often constructed by first specifying initial constraints on the general shape, and then using a fractal shape generator to fill in the surface with appropriately jagged (or smooth) details. The description employed in such graphics systems, therefore, is exactly a constrained-chance description: important details are specified, and everything else is left unspecified except in a qualitative manner.

This type of description bears a close relationship to surface interpolation methods (e.g., [24]). Typically, such schemes fit a smooth surface that satisfies whatever boundary conditions are available. The initial boundary conditions, together with the interpolation function, constitute a precise description of

*The term "representation" will be used to refer to the scheme for representing shapes, while the term "description" will be reserved for specific instances. Thus, one can compute a description of some object; it will be a member of the class of shapes that can be accounted for within the representation.

the surface shape. Such schemes are limited to smooth surfaces, however, and therefore are incapable of dealing with most natural shapes. In contrast, a fractal-based representation allows either rough or smooth surfaces to be fit to the initial boundary conditions, depending upon the fractal dimension. This method of description, therefore, is quite capable of describing most natural surfaces — and that is why the graphics community is turning to the use of fractal-based descriptions for natural surfaces.

In order to make use of this type of description it is necessary to be able to specify the surface shape in a *qualitative* manner, i.e., how rugged is the topography? This specification of qualitative shape can be accomplished by fixing the fractal dimension. The fact that we have recently developed a method of inferring the fractal dimension of the 3-D surface directly from the image data means that we are now able, for the first time, to actually compute a fractal or constrained-chance description of a real scene from its image.

Not only terrestrial topography has been modeled by use of a constrained-chance representation, but also clouds, ponds, riverbeds, snowflakes, ocean surf and stars, just to name a few examples [1,3,4,5,6,7]. Researchers have also used constrained-chance generators to produce plant shapes [1,4,6]. A very natural-looking tree can be produced by recursively applying a random number generator and simple constraints on branching geometry. In each case a random number generator plus a surprisingly small number of constraints can be used to produce very good models of apparently complex natural phenomena. Thus, there is hope for extending this approach well beyond the domain of land topography.

4.2 An Example Of Computing A Description

Figure 4 illustrates an actual example of computing such a description. Figure 4(a) is an image of a real mountain. Let us suppose that we wished to use the image data to construct a three-dimensional model of the rightmost peak (arrow), perhaps for the purpose of predicting whether or not we could climb it. I will take the standard fractal technology used in the computer graphics community as the unconstrained "primal" shape generator, as it provides an apparently accurate model of a wide range of natural surfaces.

All that is necessary to construct a description of this mountain peak is to extract shape constraints from the image and insert them into the primal shape generator. The fractal dimension of the 3-D surface is the principal parameter (constraint) required by our fractal shape generator; roughly speaking, it determines the ruggedness of the surface. The fractal dimension of the 3-D surface in the region near the rightmost peak was inferred from the fractal dimension of the image intensity surface in that area [19]. Constraint on the general outline of this peak was derived from distinguished points (those with high curvature) along the boundary between sky and mountain. These two constraints, together with the shape generator, *are* a 3-D representation of this peak; the question is: how good a representation? A view of a 3-D model derived from this representation is shown in Figure 4(b). It appears that these

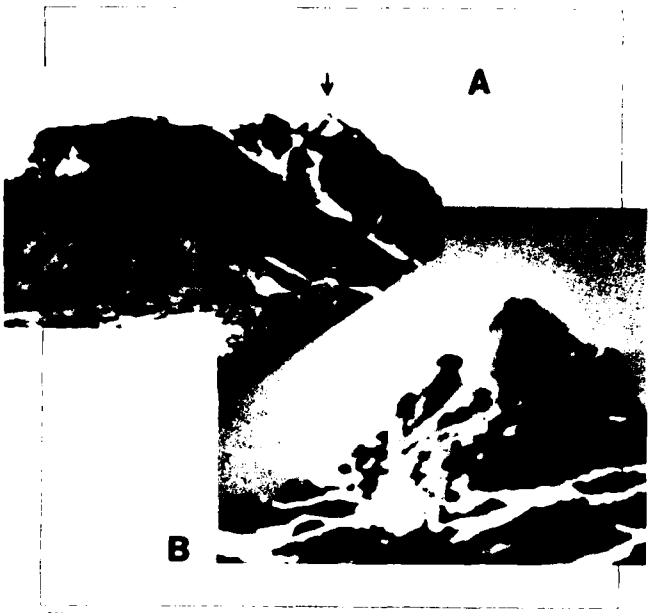


Figure 4. An example of computing a constrained-chance description.

simple constraints are sufficient for computing a good* 3-D representation of the peak.

4.3 What Do We Accomplish With This Approach?

Let's consider the problems cited above:

(1) The problem of *representing* a *complex* shape, such as a crumpled newspaper. The problem with a shape-primitive representation such as surface normals, voxcels or generalized cylinders is that the resulting description seems hopelessly complex. Because the constrained-chance representation allows us to deal only with the structural regularities and to ignore inconsequential details, the problem can become much simpler. Thus, for instance, the graphics community has found that constrained-chance fractal descriptions of complex objects (e.g., a mountain) are quite compact and easy to manipulate. It also turns out that many previously simple things, such as describing a smooth plane, remain simple.

How does this representation function when we want to compute a description of a *specific* mountain, bush or other entity from its image? Current "shape-from-x" research furnishes constraints on shape in a variety of forms: surface orientation (from texture [15 — 18,25], shading [22,23,26]), relative depth (from motion [27,28], contour [29 — 31]), and absolute depth (from stereo [32 — 34], egomotion [35,36]). It appears to be fairly straightforward to mix each of the various flavors of constraint into the vanilla-flavor shape generator [3,5], although significant research remains to be done. As more shape constraints are obtained from the image, the description becomes more and more precise; i.e., there is less and less chance in the description.

*Rather primitive ray tracing, etc., was used to generate this image; better code is being implemented.

Eventually, only one shape satisfies all of the constraints.

How complex could such a description become? The constrained-chance representation would *at worst* be as complex as a two-dimensional array of z values representing the same surface, because we could always use it to actually generate such an array of z values. As mentioned previously, experiments in human perception indicate that our representations are usually not accurate enough to recover every z value. The representation of a particular object, therefore, is likely to be quite a bit simpler than a full depth map.

(2) The problem of representing *classes* of shapes, such as are referred to by the terms "a mountain," or "a bush." Again, the ability to specify important structural details and leave the rest only qualitatively constrained allows simplification of the problem. The definition of "a mountain," for instance, might reasonably consist entirely of a specification of the fractal dimension of the surface and a caveat concerning size. If we are to judge by the results reported in the computer graphics literature, the notion of representation by constrained chance thus allows us, using only a few lines of code, to produce an accurate description of the *class* of shapes we label "mountains," or "bush."

(3) The problem of determining the set of appropriate descriptions when the shape is underconstrained by the sense data. The problem with standard shape-primitive representations is that either we must generate all combinations of shape primitives consistent with the sense data (a very hard problem), or pick a prototype and specify error bounds. The problem with using prototypes plus error bounds is that we are forced to overcommit ourselves by choosing the prototype; e.g., there is something seriously wrong about describing a cube as "a sphere $\pm 0.4r$", even though the cube certainly fits within the specified volume.

Because the constrained-chance representation allows details to be left constrained but unspecified, it allows us to deal with insufficient sense data by simply adding in those constraints that can be deduced from the image data and committing ourselves no further. The result is a programlike description that can be analyzed and manipulated, does not overcommit itself as to object shape, and allows examples of shapes consistent with the image data to be generated and examined.

(4) The problem of determining that a specific description is a member of a more general class. Here the problem with shape-primitive representations is that there is so much variability among the descriptions of the members of a class such as "mountain" that a description of the class as a whole seems extremely difficult, and determination of class membership even more so.

The problem of establishing class membership by using constrained-chance representations reduces to determining whether the constraints used to specify a particular description are a subset of those of the more general class. A determination regarding class membership is, therefore, exactly equivalent to determining whether one program's output is a subset of another program's output. While such automatic proof is a difficult problem, it is at least tractable and well-defined — unlike the equivalent problem can be when using a shape-primitive representation. Thus, a constrained-chance representation allows a clear and potentially useful definition of what it means to "recognize that x is an y."

Further, because we need only deal with the structural regularities, this problem can become much simpler than it might at first appear. Taking the class "a mountain" to be defined by fractal dimension and overall size (a definition that is actually sufficient to produce realistic mountain shapes) we can, for instance, easily determine that the description computed by us for the mountain peak is in fact a description of part of a mountain -- a task that previously seemed to be nearly impossible.

5. SUMMARY

Fractal functions seem to provide a good model of natural surface shapes. Many basic physical processes produce fractal surfaces. Fractal surfaces also *look* like natural surfaces, and so have come into widespread uses in the computer graphics community. Furthermore, we have conducted a survey of natural imagery and found that a fractal model of imaged 3-D surfaces furnishes an accurate description of both textured and shaded image regions.

Fractal functions, therefore, are useful for addressing the related problems of representing complex natural shapes such as mountains, and computing a description of such shapes from image data. The following describes the progress achieved toward the solution of these problems.

Computing a description. Characterization of image texture by means of a fractal surface model has shed considerable light on the physical basis for several of the texture techniques currently in use, and made it possible to describe image texture in a manner that is stable over transformations of scale and linear transforms of intensity. These properties of the fractal surface model allow it to serve as the basis for an accurate image segmentation procedure that is stable over a wide range of scales.

Because fractal dimension is not affected by projection distortion, its measurement can significantly enhance our ability to estimate shape from (unfamiliar) texture. Specifically, it seems that measurement of fractal dimension can provide (1) evidence of surface texture anisotropy, and (2) an estimate of the perspective texture gradient. Both capabilities are extremely important because they provide a way to obtain independent confirmation of the assumptions on which previously-reported [18] shape-from-unfamiliar-texture techniques are based.

Representing natural shapes. A constrained-chance representation modeled after the fractal techniques used by the graphics community seems useful for representing complex natural shapes, such as a crumpled newspaper or a mountain. The problem encountered when using conventional shape-primitive representations to describe natural surfaces is that the resulting description is often hopelessly complex. Because the constrained-chance representation allows us to deal only with the structural regularities and to ignore inconsequential details, the problem can become much simpler. Thus, for instance, the graphics community has found that constrained-chance fractal descriptions of complex objects (e.g., a mountain) are quite compact and easy to manipulate. Similarly, the problem of repre-

senting *classes* of shapes, such as are referred to by the terms "a mountain," or "a bush," can also be significantly simplified.

The encouraging progress that has already been achieved on both of these problems augers well for this approach. It appears that a constrained-chance representation incorporating a fractal model of surface shape will provide an elegant solution for some of the most difficult problems encountered when attempting to progress from the image of a natural scene to its description.

REFERENCES

[1] B. B. Mandelbrot, "Fractals: Form, Chance and Dimension," W. H. Freeman and Co.,San Francisco, California, 1977.

[2] L. F. Richardson, "The Problem of Contiguity: an Appendix of Statistics of Deadly Quarrels," General Systems Yearbook, vol. 6, pp. 139-187, 1961.

[3] A. Fournier, D. Fussel and L. Carpenter, "Computer Rendering of Stochastic Models," Communications of the ACM, vol. 25, 6, pp. 371-384, 1982.

[4] B. B. Mandelbrot, "The Fractal Geometry of Nature," Freeman, San Francisco, 1982.

[5] A. Norton, "Generation and Display of Geometric Fractals in 3-D," Computer Graphics, vol. 16, 3, pp. 61-67, 1982.

[6] Y. Kawaguchi, "A Morphological Study Of The Form Of Nature," Computer Graphics, vol. 16, 3, pp. 223-232, 1982.

[7] L. C. Carpenter, "Vol Libre," Computer Generated Movie, 1980.

[8] P. Brodatz, "Textures: A Photographic Album for Artists and Designers," Dover, New York, New York, 1966.

[9] K. Laws, "Textured Image Segmentation," Report 940, USC Image Processing Institute, Los Angeles, California, 1980.

[10] D. H. Ballard and C. M. Brown, "Computer Vision," Prentice-Hall Inc., Englewood Cliffs, N.J., 1982.

[11] W. K. Pratt, O.D. Faugeras, and A. Gagalowicz, "Visual Discrimination of Stochastic Texture," IEEE Transactions on Systems, Man and Cybernetics, vol SMC-8, pp. 460-473, 1978.

[12] K. Deguchi and I. Morishita, "Texture Characterization and Texture-Based Image Partitioning Using Two-Dimensional Linear Estimation Techniques," IEEE Transactions on Computers, vol C-27, pp. 739-745, 1978.

[13] A. Rosenfeld and E. B. Troy, "Visual Texture Analysis," IEEE Conference on Feature Extraction and Analysis, pp. 115-124, Argonne, Ill. Oct 1970.

[14] R. M. Haralick, K. Shanmugam and J. Dinstein, "Textural Features for Image Classification," IEEE Transactions on Systems, Man and Cybernetics, vol. SMC-3, pp. 610-621, 1973.

[15] R. Bajcsy and L. Lieberman, "Computer Description of Real Outdoor Scenes," Proceedings of 2d International Joint Conference on Pattern Recognition, pp. 174-179, Copenhagen, Aug 1974.

[16] H. Maurer, "Texture Analysis With Fourier Series," Proceedings of the 9th International Symposium on Remote Sensing of the Environment, pp 1411-1420, Ann Arbor, Michigan, April 1974.

[17] R. Bajcsy and L. Lieberman, "Texture Gradient as a Depth Cue," Computer Graphics and Image Processing, vol. 5, 1, pp 52-67, 1976.

[18] A. P. Witkin, "Recovering Surface Shape and Orientation from Texture," Artificial Intelligence, 17, pp. 17-47 (1981).

[19] A. Pentland, "Fractal Textures," Proceedings of IJCAI 83, Karlsruhe, Germany, August 1983.

[20] R. L. Gregory, "Eye And Brain: The Psychology of Seeing," New York, McGraw-Hill, 1972.

[21] D. A. Norman, "Memory and Attention," New York, Wiley, 1976.

[22] B. K. P. H. Horn, "Shape From Shading: A Method for Obtaining the Shape of a Smooth Opaque Object from One View," A.I. Technical Report 79, Project MAC, M.I.T. (1970).

[23] B. K. P. H. Horn and K. Ikeuchi, "Numerical Shape from Shading and Occluding Boundaries," Artificial Intelligence, 15, Special Issue on Computer Vision, pp. 141-184 (1981).

[24] W. E. L. Grimson, "Computing Shape Using A Theory Of Human Stereo Vision," Ph.D. Thesis, Dept. of Mathematics, M.I.T. (1980).

[25] J. R. Kender, "Shape From Texture: An Aggregation Transform that Maps a Class of Textures Into Surface Orientation," Proceedings of the Sixth International Joint Conference on Artificial Intelligence, Tokyo, Japan (1979).

[26] A. P. Pentland, "Local Computation Of Shape," Proceedings of the National Conference on Artificial Intelligence, Pittsburgh, Pennsylvania (1982).

[27] S. Ullman, "The Interpretation of Visual Motion," M.I.T. Press, Cambridge, Massachusetts (1979).

[28] D. Hoffman and B. E. Finchbaugh, "The Interpretation of Biological Motion," Bio. Cybern. 42, pp. 195-204, 1982.

[29] S. W. Zucker, R. A. Hummel, and A. Rosenfeld, "An Application of Relaxation Labeling to Line and Curve Enhancement," IEEE Transactions on Computers, C-26, 4, pp. 394-403 (1977).

[30] D. Marr, "Analysis Of Occluding Contour," Proc. Royal Soc. Lond. 197, pp. 441-475, 1977.

[31] A. P. Witkin, "Computational Theory of Line Drawing Interpretation," Artificial Intelligence Center, SRI International, Menlo Park, California (October 1981).

[32] D. C. Marr and T. Poggio, "A Computational Theory of Human Stereo Vision," Proc. R. Soc. London, 204, B, pp. 301-328 (1979).

[33] H. P. Moravec, "Rover Visual Obstacle Avoidance," Proceedings of the Seventh Joint Conference on Artificial Intelligence, pp. 785-790 (1981).

[34] H. Baker and T. O. Binford, "Depth From Edge and Intensity Based Stereo," Proceedings of the Seventh Joint Conference on Artificial Intelligence, pp. 631-636 (1981).

[35] A. Bruss and B.K.P.H. Horn, "Passive Navigation," Proceeding of the Image Understanding Workshop, Stanford, California, September 1982.

[36] K. Prazdny,"Egomotion and Relative Depth Map from Optical Flow," Technical Memo, Computer Science Department, University of Essex, Colchester, England.

Appendix D

PERCEPTUAL ORGANIZATION AND CURVE PARTITIONING

By: M.A. Fischler and R.C. Bolles

PERCEPTUAL ORGANIZATION AND CURVE PARTITIONING*

By: M.A. Fischler and R.C. Bolles

SRI International
333 Ravenswood Avenue
Menlo Park, California 94025

ABSTRACT

In this paper we offer a critical evaluation of the partitioning (perceptual organization) problem, noting the extent to which it has distinct formulations and parameterizations. We show that most partitioning techniques can be characterized as variations of four distinct paradigms, and argue that any effective technique must satisfy two general principles. We give concrete substance to our general discussion by introducing new partitioning techniques for planar geometric curves, and present experimental results demonstrating their effectiveness.

I INTRODUCTION

A basic attribute of the human visual system is its ability to group elements of a perceived scene or visual field into meaningful or coherent clusters; in addition to clustering or partitioning, the visual system generally imparts structure and often a semantic interpretation to the data. In spite of the apparent existence proof provided by human vision, the general problem of scene partitioning remains unsolved for computer vision. Furthermore, there is even some question as to whether this problem is meaningful (or a solution verifiable) in its most general form.

Part of the difficulty resides in the fact that it is not clear to what extent semantic knowledge (e.g., recognizing the appearance of a straight line or some letter of the English alphabet), as opposed to generic criteria (e.g., grouping scene elements on the basis of geometric proximity), is employed in examples of human performance. It would not be unreasonable to assume that a typical human has on the order of tens of thousands of iconic primitives in his visual vocabulary; a normal adult's linguistic vocabulary might consist of from 10,000 to 40,000 root words, and iconic memory is believed to be at *least as effective as* its linguistic counterpart. Since, at present, we cannot hope to duplicate human competence in semantic interpretation, it would be desirable to find a task domain in which the influence of semantic knowledge is limited. In such a domain it might be possible to discover the generic criteria employed by the human visual system and to duplicate human performance. One of the main goals of the research effort described in this paper is to find a set of generic rules and models that will permit a machine to duplicate human performance in partitioning planar curves.

II THE PARTITIONING PROBLEM: ISSUES AND CONSIDERATIONS

Even if we are given a problem domain in which explicit semantic cues are missing, to what extent is partitioning dependent on the purpose, vocabulary, data representation, and past experience of the "partitioning instrument," as opposed to being a search for context independent "intrinsic structure" in the data? We argue that rather than having a unique formulation, the partitioning problem must be paramaterized along a number of basic dimensions. In the remainder of this section we enumerate some of these dimensions and discuss their relevance.

A. Intent (Purpose) of the Partitioning Task

In the experiment described in Figure 1, human subjects were presented with the task of partitioning a set of two-dimensional curves with respect to three different objectives: (1) choose a set of contour points that best mark those locations at which curve segments produced by different processes were "glued" together; (2) choose a set of contour points that best allow one to reconstruct the complete curve; (3) choose a set of contour points that would best allow one to distinguish the given curve from others. Each person was given only one of the three task statements. Even though the point selections within a task varied from subject to subject, there was significant overlap and the variations were easily explained in terms of recognized strategies invoked to satisfy the given constraints; however, the points selected in the three tasks were significantly different. Thus, even in the case of data with almost no semantic content, the partitioning problem is NOT a generic task independent of purpose.

---

* The research reported herein was supported by the Defense Advanced Research Projects Agency under Contract No. MDA 903-83-C-0027 and by the National Science Foundation under Contract No ECS-7917028.

B. Partitioning Viewed as an Explanation of Curve Construction

With respect to "process partitioning" (partitioning the curve into segments produced by different processes), a partition can be viewed as an explanation of how the curve was constructed. Explanations have the following attributes which, when assigned different "values," lead to different explanations and thus different partitions:

* Vocabulary (primitives and relations) -- what properties of our data should be represented, and how should these properties be computed? That is, we must select those aspects of the problem domain we consider relevant to our partition decisions (e.g., geometric shape, gray scale, line width, semantic content), and enable their computation by providing models for the corresponding structures (e.g., straight-line segment, circular arc, wiggly segment). We must also allow for the appropriate "viewing" conditions; e.g., symmetry, repeated structure, parallel lines, are global concepts that imply that the curve has finite extent and can be viewed as a "whole," as opposed to only permitting computations that are based on some limited interval or neighborhood of (or along) the curve.

* Definition of Noise -- in a generic sense, any data set that does not have a "simple (concise)" description is noise. Thus, noise is relative to both the selected descriptive language and an arbitrary level of complexity. The particular choices for vocabulary and the acceptable complexity level determine whether a point is selected as a partition point or considered to be a noise element.

* Believability -- depending on the competence (completeness) of our vocabulary to describe any curve that may be encountered, the selected metric for judging similarity, and the arbitrary threshold we have chosen for believing that a vocabulary term corresponds to some segment of a given curve, partition points will appear, disappear, or shift.

C. Representation

The form in which the data is presented (i.e., the input representation), as well as the type of data, are critical aspects of the problem definition, and will have a major impact on the decisions made by different approaches to the partitioning task. Some of the key variables are:

* Analog (pictorial) vs digital (quantized) vs analytic description of the curves

* Single vs multiple "views" (e.g., single vs. multiple quantizations of a given segment)

* Input resolution vs. length of smallest segment of interest

* Simply-connected (continuous) curves vs self-intersecting curves or curves with "gaps"

* For complex situations, is connectivity provided, or must it be established

* If a curve possesses attributes (e.g., gray scale, width) other than "shape" that are to serve as partitioning criteria, how are they obtained -- by measurement on an actual "image," or as symbolic tags provided as part of the given data set?

D. Evaluation

How do we determine if a given technique or approach to the partitioning problem is successful? How can we compare different techniques? We have already observed that, to the extent that partitioning is a "well-defined" problem at all, it has a large number of alternative formulations and parameterizations. Thus, a technique that is dominant under one set of conditions may be inferior under a different parameterization. Never the less, any evaluation procedure must be based on the following considerations:

* Is there a known "correct" answer (e.g., because of the way the curves were constructed)?

* Is the problem formulated in such a way that there is a "provably" correct answer?

* How good is the agreement of the partitioned data with the descriptive vocabulary (models) in which the "explanation" is posed?

* How good is the agreement with (generic or "expert") subjective human judgment?

* What is the trade-off between "false-alarms" and "misses" in the placement of partition points. To the extent that it is not possible to ensure a perfect answer (in the placement of the partition points), there is no way to avoid such a trade-off. Even if the the relative weighting between these two types of errors is not made explicit, it is inherent in any decision procedure -- including the use of subjective human judgment.

In spite of all of the previous discussion in this section, it might still be argued that if we take the union of all partition points obtained for all reasonable definitions and parameterizations of the partition problem, we would still end up with a "small" set of partition points for any given curve, and further, there may be a generic procedure for obtaining this covering set. While a full discussion of this possibility is is not feasible here, we can construct a counterexample to the unqualified conjecture based on selecting a very high ratio of the cost of a miss to a false-alarm in selecting the partition points. A (weak) refutation can also be based on the observation that if a generic covering set of partition points exists, then there should be a relatively consistent way of ordering all the points on a given curve as to their being acceptable partition

points; the experiment presented in Figure 1 indicates that, in general, such a consistent ordering does not exist.

III PARADIGMS FOR CURVE PARTITIONING

Almost all algorithms employed for curve partitioning appear to be special cases (instantiations) of one or more of the following paradigms:

* Local Detection of Distinguished Points: a partition point is inserted at locations along the curve at which one or more of the descriptive attributes (e.g., curvature, distance from a coordinate axis or centroid) is determined to have a discontinuity, an extreme value (maxima or minima), or a zero value separating intervals of positive and negative values.
* Best Global Description: a set of partition points is inserted at those locations along a curve that allow the "best" description of the associated segments in terms of some a priori set of models (e.g., the set of models might consist of all first and second degree polynomials, with only one model permitted to explain the data between two adjacent partition points; the quality of the description might be measured by the mean square deviation of the data points from the fitting polynomials).
* Confirming Evidence: given a number of "independent" procedures (or possibly different parameterizations of a given procedure) for locating potential partition points, we retain only those partition points that are common to some subset of the different procedures or their parameterizations.
* Recursive Simplification: the input data is subjected to repeated applications of some transformation that monotonically reduces some measurable aspect of the data to one of a finite number of terminal states (e.g., differentiation, smoothing, projection, thresholding). The hierarchy of data sets thus produced is then processed with an algorithm derived from the previous three paradigms.

IV PRINCIPLES OF EFFECTIVE (ROBUST) MODEL-BASED INTERPRETATION

What underlies our choice of partitioning criteria? We assert that any competent partitioning technique, regardless of which of the above paradigms is employed, will incorporate the following principles.

A. Stability

The "principle of stability," is the assertion that any valid perceptual decision should be stable under at least small perturbations of both the imaging conditions and the decision algorithm parameters. This generalization of the assumption of "general position" also subsumes the assertion (often presented as an assumption) that most of a scene must be describable in terms of continuous variables if meaningful interpretation is to be possible.

It is interesting to observe that many of the constructs in mathematics (e.g., the derivative) are based on the concepts of convergence and limit, also subsumed under the stability principle. Attempts to measure the digital counterparts of the mathematical concepts have traditionally employed window type "operators" that are not based on a limiting process; it should come as no surprise that such attempts have not been very effective.

In practice, if we perturb the various imaging and decision parameters, we observe relatively stable decision regions separated by obviously unstable intervals (e.g., the two distinct percepts produced by a Necker cube). The stable regions represent alternative hypotheses that generally cannot be resolved without recourse to either additional and more restrictive assumptions, or semantic (domain-specific) knowledge.

B. Complete, Concise, and Complexity Limited Explanation

The decision-making process in image interpretation, i.e. matching image derived data to a priori models, not only must be stable, but must also explain all the structure observable in the data. Equally important, the explanation must satisfy specific criteria for believability and complexity. Believability is largely a matter of offering the simplest possible description of the data and, in addition, explaining any deviation of the data from the models (vocabulary) used in the description. Even the simplest description, however, must also be of limited complexity; otherwise or it will not be understandable and thus not believable.

By making the foregoing principles explicit, we can directly invoke them (as demonstrated in the following section) to formulate effective algorithms for perceptual organization.

V INSTANTIATION OF THE THEORY: SPECIFIC TECHNIQUES FOR CURVE PARTITIONING

In this section we offer two effective new algorithms for curve partitioning (program listings available from the authors). In each case, we first describe the the algorithm, and later indicate how it was motivated and constrained by the principles just presented. In both algorithms, the key ideas are: (1) to view each point, or segment of a curve, from as many perspectives as possible, retaining only those partition points

receiving the highest level of multiple confirmation; and (2) inhibiting the further selection of partition points when the density of points already selected exceeds a preselected or computed limit.

A. Curve Partitioning Based on Detecting Local Discontinuity

In this sub-section we present a new approach to the problem of finding points of discontinuity ("critical points") on a curve. Our criterion for success is whether we can match the performance of human subjects given the same task (e.g., see Figure 1). The importance of this problem from the standpoint of the psychology of human vision dates back to the work of Attneave [1954]. However, it has long been recognized as a very difficult problem, and no satisfactory computer algorithm currently exists for this purpose. An excellent discussion of the problem may be found in in Davis [1977]; other pertinent references include Rosenfeld [1975], Freeman [1977], Kruse [1978], and Pavlidis [1980]. Results and observations akin and complementary to those presented here can be found in Hoffman [1982] and in Witkin [1983].

Most approaches equate the search for critical points with looking for points of high curvature. Although this intuition seems to be correct, it is incomplete as stated (i.e., it does not explicitly take into account "explanation" complexity); further, the methods proposed for measuring curvature are often inadequate in their selection of stability criteria. In Figure 2 we show some results of measuring curvature using discrete approximations to the mathematical definition.

We have developed an algorithm for locating critical points that invokes a model related to, but distinct from, the mathematical concept of curvature. The algorithm labels each point on a curve as belonging to one of three categories: (a) a point in a smooth interval, (b) a critical point, or (c) a point in a noisy interval. To make this choice, the algorithm analyzes the deviations of the curve from a chord or "stick" that is iteratively advanced along the curve (this will be done for a variety of lengths, which is analogous to analyzing the curve at different resolutions). If the curve stays close to the chord, points in the interval spanned by the chord will be labeled as belonging to a smooth section. If the curve makes a single excursion away from the chord, the point in the interval that is farthest from the chord will be labeled a critical point (actually, for each placement of the chord, an accumulator associated with the farthest point will be incremented by the distance between the point and the chord). If the curve makes two or more excursions, points in the interval will be labeled as noise points.

We should note here that "noisy" intervals at low resolution (large chord length) will have many critical points at higher resolution (small chord length). Figure 3 shows examples of curve segments and their classifications. The distance from a chord that defines a significant excursion (i.e., the width of the boxes in Figure 3) is a function of the expected noise along the curve and the length of the chord.

At each resolution (i.e., stick size), the algorithm orders the critical points according to the values in their accumulators and selects the best ones first. To avoid setting an arbitrary "goodness" threshold for distinguishing critical from ordinary points, we use a complexity criterion. To halt the selection process, we stop when the points being suggested are too close to those selected previously at the given resolution. In our experiments we define "too close" as being within a quarter of the stick length used to suggest the point.

After the critical points have been selected at the coarsest resolution, the algorithm is applied at higher resolutions to locate additional critical points that are outside the regions dominated by previously selected points. Figure 4a shows the critical points determined at the coarsest level (stick length of 100 pixels; approximately 1/10 of the length of the curve). Figure 4b shows all the critical points labeled with the stick lengths used to determine them. (We note that this critical point detection procedure does not locate inflection points or smooth transitions between segments, such as the transition from an arc of a circle to a line tangent to the circle.)

The above algorithm appears to be very effective, especially for finding obvious partition points and in not making "ugly" mistakes (i.e., choosing partition points at locations that none of our human subjects would pick). Its ability to find good partition points is based on evaluating each point on the curve from multiple viewpoints (placements of the stick) -- a direct application of the principle of stability. Requiring that the partition points remain stable under changes in resolution (i.e., small changes in stick length) did not appear to be effective and was not employed; in fact, stick length was altered by a significant amount in each iteration, and partition points found at these different scales of resolution were not expected to support each other, but were assumed to be due to distinct phenomena.

The avoidance of ugly mistakes was due to our method of limiting the number of partition points that could be selected at any level of resolution, or in any neighborhood of a selected point (i.e., limiting the explanation complexity). One concept we invoked here, related to that of complete explanation, was that the detection procedure could not be trusted to provide an adequate explanation when more than a single critical point was in its field of view, and in such a situation, any decision was deferred to later iterations at higher levels of resolution (i.e., shorter stick lengths).

Finally, in accord with our previous discussion, the algorithm has two free parameters that provide control over its definition of noise (i.e., variations too small or too close together to be of interest), and its willingness to miss a good partition point so as to be sure it does not select a bad one.

B. Curve Partitioning Based and Detecting Process Homogenity

To match human performance in partitioning a curve, by recognizing those locations at which one generating process terminates and another begins, is orders of magnitude more difficult than partitioning based on local discontinuity analysis. As noted earlier, a critical aspect of such performance is the size and effectiveness of the vocabulary (of a priori models) employed. Explicitly providing a general purpose vocabulary to the machine would entail an unreasonably large amount of work -- we hypothesize that the only effective way of allowing a machine to acquire such knowledge is to provide it with a learning capability.

For our purposes in this investigation, we chose a problem in which the relevant vocabulary was extremely limited: the curves to be partitioned are composed exclusively of straight lines and arcs of circles. (Two specific applications we were interested in here were the decomposition of silhouettes of industrial parts, and the decomposition of the line scans returned by a "structured light" ranging device viewing scenes containing various diameter cylinders and planar faced objects lying on a flat surface.) Our goal here was to develop a procedure for locating critical points along a curve in such a way that the segments between the critical points would be satisfactorily modeled by either a straight-line segment or a circular arc. Relevant work addressing this problem has been done by Montanari [1970], Ramer [1972], Pavlidis [1974], Liao [1981], and Lowe [1982].

Our approach is to analyze several "views" of a curve, construct a list of possible critical points, and then select the optimum points between which models from our vocabulary can be fitted. For our experiments we quantized an analytic curve at several positions and orientations (with respect to a pixel grid), then attempted to recover the original model.

For each view (quantization) of the curve we locate occurrences of lines and arcs, marking their ends as prospective partition points. This is accomplished by randomly selecting small seed segments from the curve, fitting to them a line or arc, examining the fit, and then extending as far as possible those models that exhibit a good fit. After a large number of seeds have been explored in the different views of the curve, the histogram (frequency count as a function of path length) of beginnings and endings is used to suggest critical points (in order of their frequency of occurrence). Each new critical point, considered for inclusion in the explanation of how the curve is constructed, introduces two new segments which are compared to both our line and circle models. If one or both of the segments have acceptable fits, the corresponding curve segments are marked as explained. Otherwise, the segments are left to be explained by additional critical points and the partitions they imply. The addition of critical points continues until the complete curve is explained. Figure 5 shows an example of the operation of this algorithm.

While admittedly operating in a relatively simple environment, the above algorithm exhibits excellent performance. This is true even in the difficult case of finding partition points along the smooth interface between a straight line and a circle to which the line is tangent.

Both basic principles, stability and complete explanation, are deeply embedded in this algorithm. Retaining only those partition points which persist under different "viewpoints" was motivated by the principle of stability. Our technique for evaluating the fit of the segment of a curve between two partition points, to both the line and circle models, requires that the deviations from an acceptable model have the characteristics of "white" (random) noise; this is an instantiation of the principle of complete explanation, and is based on our previous work presented in Bolles [1982].

VI DISCUSSION

We can summarize our key points as follows:

* The partition problem does not have a unique definition, but is parameterized with respect to such items as purpose, data representation, trade-off between different error types (false-alarms vs misses), etc.
* Psychologically acceptable partitions are associated with an implied explanation that must satisfy criteria for accuracy, complexity, and believability. These criteria can be formulated in terms of a set of principles, which, in turn, can guide the construction of effective partitioning algorithms (i.e., they provide necessary conditions).

One implication contained in these observations is that a purely mathematical definition of "intrinsic structure" (i.e., a definition justified solely by appeal to mathematical criteria or principles) cannot, by itself, be sufficiently selective to serve as a basis for duplicating human performance in the partitioning task; generic partitioning (i.e., partitioning in the absence of semantic content) is based on psychological "laws" and physiological mechanisms, as well as on correlations embedded in the data.

In this paper we have looked at a very limited subset of the class of all scene partitioning problems; nevertheless, it is interesting to speculate on how the human performs so effectively in the broader domain of interpreting single images of natural scenes. The speed of response in the humans ability to interpret a sequence of images of dissimilar scenes makes it highly questionable that there is some mechanism by which he simultaneously matches all his semantic primitives against the imaged data, even if we assume that some independent process has already presented him with a "camera model" that resolves some of the uncertainties in image scale, orientation, and projective distortion. How does the human index

B. Curve Partitioning Based and Detecting Process Homogenity

To match human performance in partitioning a curve, by recognizing those locations at which one generating process terminates and another begins, is orders of magnitude more difficult than partitioning based on local discontinuity analysis. As noted earlier, a critical aspect of such performance is the size and effectiveness of the vocabulary (of a priori models) employed. Explicitly providing a general purpose vocabulary to the machine would entail an unreasonably large amount of work -- we hypothesize that the only effective way of allowing a machine to acquire such knowledge is to provide it with a learning capability.

For our purposes in this investigation, we chose a problem in which the relevant vocabulary was extremely limited: the curves to be partitioned are composed exclusively of straight lines and arcs of circles. (Two specific applications we were interested in here were the decomposition of silhouettes of industrial parts, and the decomposition of the line scans returned by a "structured light" ranging device viewing scenes containing various diameter cylinders and planar faced objects lying on a flat surface.) Our goal here was to develop a procedure for locating critical points along a curve in such a way that the segments between the critical points would be satisfactorily modeled by either a straight-line segment or a circular arc. Relevant work addressing this problem has been done by Montanari [1970], Ramer [1972], Pavlidis [1974], Liao [1981], and Lowe [1982].

Our approach is to analyze several "views" of a curve, construct a list of possible critical points, and then select the optimum points between which models from our vocabulary can be fitted. For our experiments we quantized an analytic curve at several positions and orientations (with respect to a pixel grid), then attempted to recover the original model.

For each view (quantization) of the curve we locate occurrences of lines and arcs, marking their ends as prospective partition points. This is accomplished by randomly selecting small seed segments from the curve, fitting to them a line or arc, examining the fit, and then extending as far as possible those models that exhibit a good fit. After a large number of seeds have been explored in the different views of the curve, the histogram (frequency count as a function of path length) of beginnings and endings is used to suggest critical points (in order of their frequency of occurrence). Each new critical point, considered for inclusion in the explanation of how the curve is constructed, introduces two new segments which are compared to both our line and circle models. If one or both of the segments have acceptable fits, the corresponding curve segments are marked as explained. Otherwise, the segments are left to be explained by additional critical points and the partitions they imply. The addition of critical points continues until the complete curve is explained. Figure 5 shows an example of the operation of this algorithm.

While admittedly operating in a relatively simple environment, the above algorithm exhibits excellent performance. This is true even in the difficult case of finding partition points along the smooth interface between a straight line and a circle to which the line is tangent.

Both basic principles, stability and complete explanation, are deeply embedded in this algorithm. Retaining only those partition points which persist under different "viewpoints" was motivated by the principle of stability. Our technique for evaluating the fit of the segment of a curve between two partition points, to both the line and circle models, requires that the deviations from an acceptable model have the characteristics of "white" (random) noise; this is an instantiation of the principle of complete explanation, and is based on our previous work presented in Bolles [1982].

VI DISCUSSION

We can summarize our key points as follows:

* The partition problem does not have a unique definition, but is parameterized with respect to such items as purpose, data representation, trade-off between different error types (false-alarms vs misses), etc.
* Psychologically acceptable partitions are associated with an implied explanation that must satisfy criteria for accuracy, complexity, and believability. These criteria can be formulated in terms of a set of principles, which, in turn, can guide the construction of effective partitioning algorithms (i.e., they provide necessary conditions).

One implication contained in these observations is that a purely mathematical definition of "intrinsic structure" (i.e., a definition justified solely by appeal to mathematical criteria or principles) cannot, by itself, be sufficiently selective to serve as a basis for duplicating human performance in the partitioning task; generic partitioning (i.e., partitioning in the absence of semantic content) is based on psychological "laws" and physiological mechanisms, as well as on correlations embedded in the data.

In this paper we have looked at a very limited subset of the class of all scene partitioning problems; nevertheless, it is interesting to speculate on how the human performs so effectively in the broader domain of interpreting single images of natural scenes. The speed of response in the humans ability to interpret a sequence of images of dissimilar scenes makes it highly questionable that there is some mechanism by which he simultaneously matches all his semantic primitives against the imaged data, even if we assume that some independent process has already presented him with a "camera model" that resolves some of the uncertainties in image scale, orientation, and projective distortion. How does the human index

into the large semantic data base to find the appropriate models for the scene at hand?

Consider the following paradigm: first a set of coherent components is recovered from the image on the basis of very general (but parameterized) clustering criteria of the type described earlier; next, a relatively small set of semantic models, which are components of many of the objects in the complete semantic vocabulary, are matched against the extracted clusters; successful matches are then used to index into the full data base and the corresponding entries are matched against both the extracted clusters and adjacent scene components; these additional successful matches will now trigger both iconic and symbolic associations that result in further matching possibilities as well as perceptual hypotheses that organize large portions of the image into coherent structures (gestalt phenomena).

If this paradigm is valid, then, even though much of the perceptual process would depend on an individual's personal experience and immediate goals, we might still expect "hard wired" algorithms (genetically programmed, but with adjustable parameters) to be employed in the initial partitioning steps.

In this paper, we have attempted to give computational definitions to some of the organizing criteria needed to approach human level performance in the partitioning task. However, we believe that our more important contribution has been the explicit formulation of a set of principles that we assert must be satisfied by any effective procedure for perceptual grouping.

REFERENCES

1. Attneave, F., "Some Aspects of Visual Perception," Psychol. Rev., Vol. 61, pp. 183-193 (1954).

2. Bolles, R.C., M.A. Fischler, "A RANSAC-based Approach to Model Fitting and Its Application to Finding Cylinders in Range Data," in Proc. of the Seventh International Joint Conference on Artificial Intelligence, Vancouver, B.C., Canada, pp. 637-643 (August 1982).

3. Davis, L.S., "Understanding Shape: Angles and Sides," IEEE Transactions on Computers, Vol. C-26, pp. 236-242 (March 1977).

4. Freeman, H., L.S. Davis, "A Corner-finding Algorithm for Chain-Coded Curves," IEEE Transactions on Computers, Vol. C-26, pp. 297-303 (March 1977).

5. Hoffman, D.D., W.A. Richards, "Representing Smooth Plane Curves for Recognition: Implications for Figure-Ground Reversal," in Proc. of the Second National Conference on Artificial Intelligence, Pittsburgh, Pennsylvania, pp. 5-8 (August 1982).

6. Kruse, B., C.V.K. Rao, "A Matched Filtering Technique for Corner Detection," in Proc. of the Fourth International Joint Conference on Pattern Recognition, Kyoto, Japan, pp. 642-644 (November 1978).

7. Liao, Y., "A Two-Stage Method of Fitting Conic Arcs and Straight-Line Segments to Digitized Contours," in Proc. of the Pattern Recognition and Image Processing Conference, Dallas, Texas, pp. 224-229 (August 1981).

8. Lowe, D.G., T.G. Binford, "Segmentation and Aggregation; an Approach to Figure-Ground phenomena," Proceedings of the Image Understanding Workshop, Stanford University, Stanford, California (September 1982).

9. Montanari, U., "A Note on Minimal Length Polygonal Approximation to a Digitized Contour," Communications of the ACM, Vol. 13, pp. 41-47 (January 1970).

10. Pavlidis, T., "Algorithms for Shape Analysis of Contours and Waveforms," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-2, pp. 301-312 (July 1980).

11. Pavlidis, T., S.L. Horowitz, "Segmentation of Plane Curves," IEEE Transactions on Computers, Vol. C-23, pp. 860-870 (August 1974).

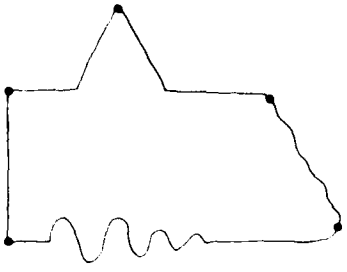
12. Ramer, U., "An Iterative Procedure for the Polygonal Approximation of Plane Curves," Computer Graphics and Image Processing, Vol. 1, pp. 244-256 (1972).

13. Rosenfeld, A., J.S. Weszka, "An Improved Method of Angle Detection on Digital Curves," IEEE Transactions on Computers, Vol. C-24, pp. 940-941 (September 1975).

14. Witkin, A., "Scale-Dependent Qualitative Signal Description," (in preparation, 1983).

TASK 1: Select *AT MOST* 5 points to describe this line drawing so that you will be able to reconstruct it as well as possible 10 years from now, given just the sequence of selected points.

Since five points were sufficient to form an approximate convex hull of the figure, virtually everyone did so, selecting the 5 points shown below.



TASK 2: Assume that a friend of yours is going to be asked to *recognize* this line drawing on the basis of the information you supply him about it. He will be presented with a set of drawings, one of which will be a rotated and scaled version of this curve. You are only allowed to provided him with *A SEQUENCE OF AT MOST 5 POINTS*. Mark the points you would select.

Since 5 points were not enough to outline all the key features of the figure, the subjects had to decide what to leave out. They seemed to adopt one of two general strategies: (a) use the limited number of points to describe one distinct feature well (illustrated by the selection on the left), or (b) use the points to outline the basic shape of the figure (shown on the right).

TASK 3: This line drawing was constructed by piecing together segments produced by different processes. Please indicate where you think the junctions between segments occur *AND VERY BRIEFLY DESCRIBE EACH SEGMENT*. Use as few points as possible, but no more than 5.

The constraint of being limited to 5 points forced the subjects to consider the whole curve and develop a consistent, global explanation. The basic strategy seemed to be a recursive one in which they first partitioned the curve into 2 segments by placing a breakpoint at position 1 and another one at either position 2 or position 3 to separate the smooth curves from the sharp corners. Then they used the remaining points to subdivide these segments according to a vocabulary they selected that included such things as triangles, rectangles, and sinusoids. For example, almost everyone placed breakpoints at positions 3 and 4 and described the enclosed segment as part of a triangle. Similarly the segment between positions 1 and 5 was generally described as a decaying sinusoid. It is interesting to note that in task 1 the subjects consistently placed a point close to position 5 but always farther to the right, because they were trying to approximate a convex hull. The different purposes led to different placements.

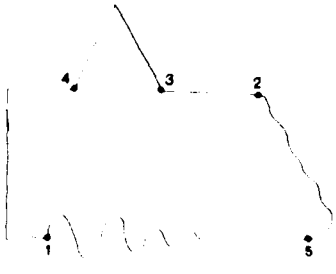
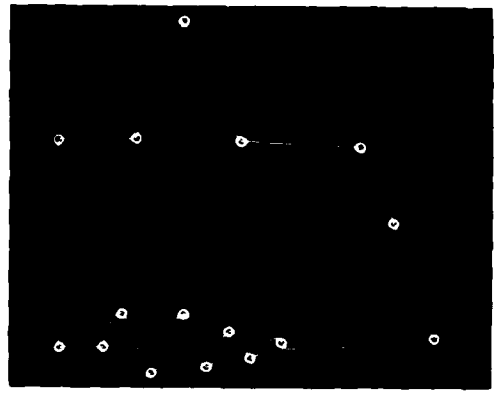
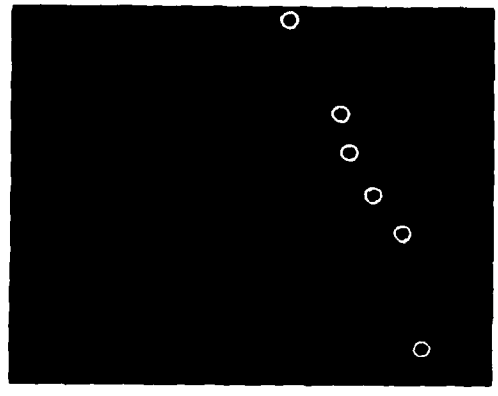


FIGURE 1 EXPERIMENTS IN WHICH HUMAN SUBJECTS WERE ASKED TO SEGMENT A CURVE



(a) This figure shows the results of applying the "improved angle detection" procedure described in Rosenfeld [1975] to a digitized version of the curve in Figure 1. The procedure works quite well, except for the introduction of a breakpoint in the middle of the right side and the merging of two small bumps at the right of the sinusoidal segment.



(b) However, if we extract a portion of the curve and apply the algorithm, it introduces several additional breakpoints because the change in curve length causes some of the algorithm parameters to change.

FIGURE 2 ESTIMATION OF CURVATURE FROM DISCRETE APPROXIMATIONS

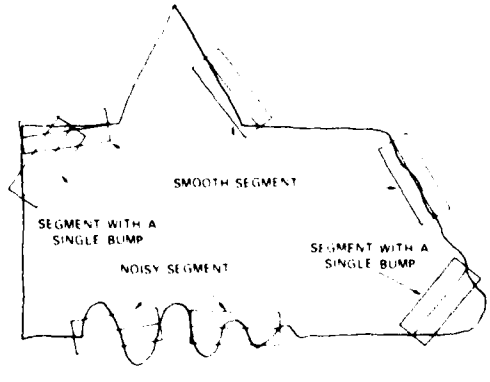
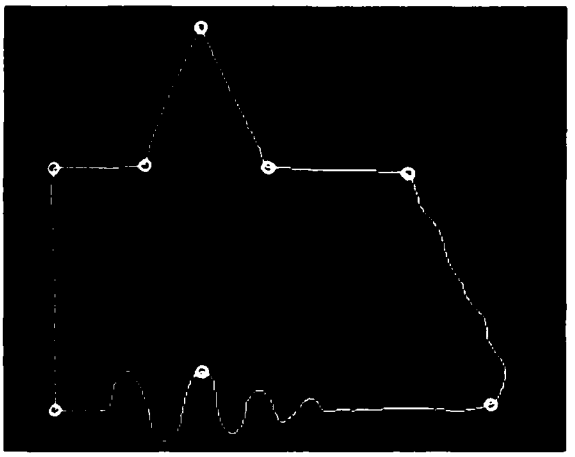
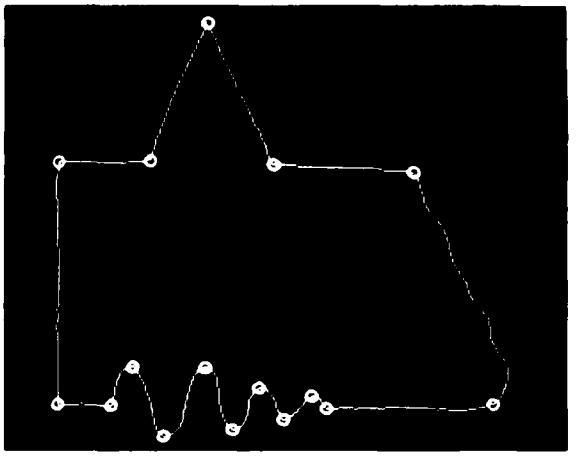


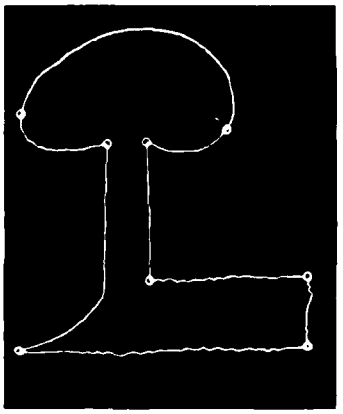
FIGURE 3 EXAMPLE CURVE SEGMENTS AND THEIR CLASSIFICATIONS

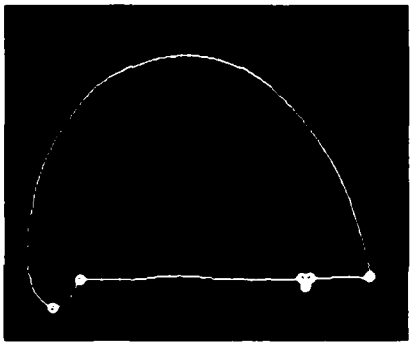


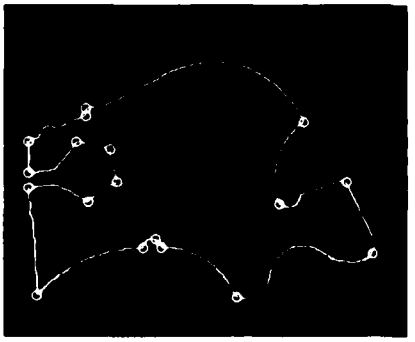
(a) Results of the analysis at the coarsest resolution (i.e., a stick length of 100 pixels) which approximately half of the lengths of the curve

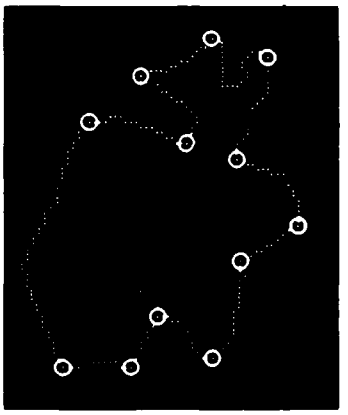


(b) Results from all resolutions (100, 80, 40, 20, 15, 10)



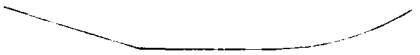




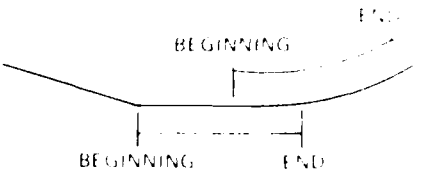


(c) Additional examples of the local discontinuity analysis

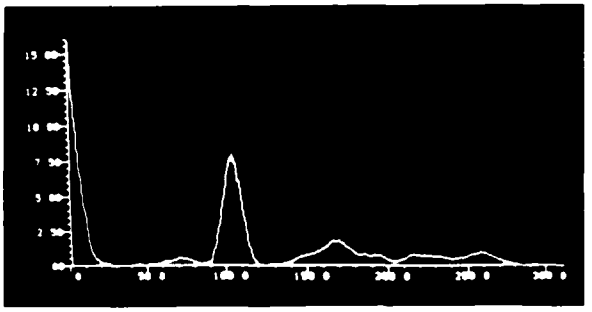
FIGURE 4 LOCAL DISCONTINUITY PARTITIONING



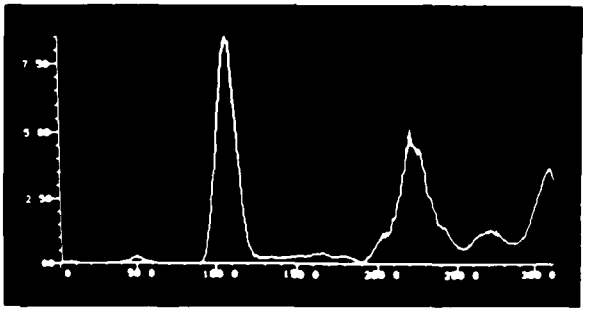
(a)



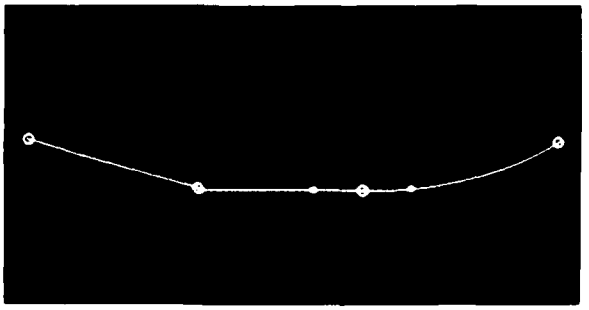
(b)



(c)

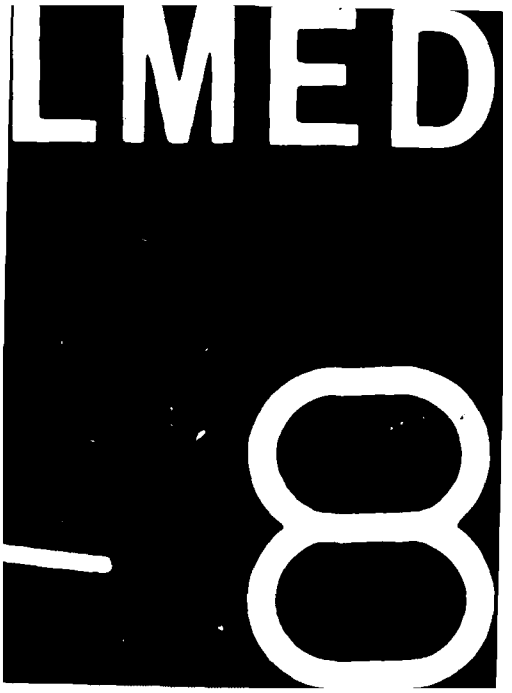


(d)



(e)

FIGURE 5 PROCESS PARTITIONING


LMED
8